INSTITUT FÜR DEUTSCHE SPRACHE

FORSCHUNGSBERICHTE

herausgegeben von

Ulrich Engel und Irmgard Vogel

Band 14

Mannheim 1974

» Tübinger Beiträge zur Linguistik «
74 Tübingen 1 · Postfach 2567
Printed in Germany
Druck: FOTODRUCK PRÄZIS B. v. Spangenberg KG · Tübingen
ISBN 3–87808–614–8

Maschinelle Textverarbeitung

im Rechenzentrum des IdS

GESAMTINHALTSVERZEICHNIS

Vorbemerkungen der Herausgeber

Seit dem Herbst 1969 verfügt das Institut für deutsche Sprache über eine mittelgroße Datenverarbeitungsanlage Siemens 4004/35. Mitte 1974 wird eine große Rechenanlage(Siemens 4004/151) installiert werden. Sie wird zunächst in erster Linie der Durchführung des Projekts "Linguistische Datenverarbeitung II" dienen, soll auf lange Sicht aber auch einer Reihe von anderen linguistischen Forschungsinstituten zur Verfügung stehen. Damit ist das Institut für deutsche Sprache dank der großzügigen Förderung durch das Bundesministerium für Forschung und Technologie das einzige linguistische Forschungsinstitut in der Bundesrepublik, das über eine eigene leistungsfähige Rechenanlage verfügt.

Dieser Erfolg ist dem IdS nicht in die Wiege gelegt worden. Es stand in der Anfangszeit nicht einmal fest, daß das Institut bei seinen Forschungen überhaupt datenverarbeitende Maschinen einsetzen würde. Der Beginn der linguistischen Datenverarbeitung im IdS liegt im Februar 1965, als die damalige neugebildete "Kommission für datenverarbeitende Maschinen und Sprachforschung" nach reger und teilweise kontroverser Diskussion beschloß, daß das Institut sich künftig auch der durch den Computer gebotenen Möglichkeiten bedienen solle.

Nach Lage der Dinge kam damals nur eine Zusammenarbeit mit dem Deutschen Rechenzentrum in Darmstadt in Frage. Dort stand eine große Rechenanlage vom Typ IBM 7090, später 7094, zur Verfügung. Vom Frühsommer 1965 an hat Gerhard Stickel - damals wissenschaftlicher Mitarbeiter des DRZ, heute Leiter der Abteilung Kontrastive Linguistik im IdS - mit Sachkunde und großem persönlichen Einsatz die Erstellung eines maschinengespeicherten Corpus für das IdS betreut. Ihm ist in erster Linie für die Bewältigung der kaum übersehbaren Anfangsschwierigkeiten und für die ersten Erfolge zu danken. Auch die wohlwollende Förderung durch Friedhelm Schulte-Tigges, den Leiter der Abteilung Nichtnumerik im DRZ, ist dankbar zu erwähnen.

Die unmittelbare Datenerfassung auf Lochstreifen erfolgte damals wie heute im IdS, seinerzeit unter unmittelbarer Beteiligung fast sämtlicher wissenschaftlicher Mitarbeiter, die sich mit Ablochvorschriften, Korrekturen usw. herumzuschlagen hatten. Erst im Jahre 1966 war es möglich, einen wissenschaftlichen Mitarbeiter eigens für die Texterfassung einzustellen. Ingeborg Zint hat unter großen persönlichen Opfern die schwierigen und immer umfangreicheren Arbeiten am DRZ nach dem Weggang von Gerhard Stickel durchgeführt. Ihr trat ein Jahr später Paul Wolfangel zur Seite, der, nachdem Frau Zint um eine Rückversetzung an die Mannheimer Zentrale gebeten hatte, zeitweise allein, dann mit Unterstützung anderer Mitarbeiter die Arbeiten weiterführte. Heute verfügt das IdS über ein gut eingearbeitetes Team von Datenverarbeitungsfachleuten und Schreibkräften. Die Erweiterung und Verbesserung des "Mannheimer Corpus", mit der seit einiger Zeit Pantelis Nikitopoulos und einige seiner Mitarbeiter befaßt sind, wird von diesem Team bewältigt werden können.

Parallel zu den Anfängen der Texterfassung standen Überlegungen, wie man den Computer auf andere Weise für linguistische Untersuchungen verwenden könne. Der Klartext liefert ja erst dann die erforderlichen Materialien für linguistische Untersuchungen, wenn ein vollständig funktionierendes Analyseverfahren zur Verfügung steht. Davon ist man auch heute noch weit entfernt, trotz erfolgreicher Bemühungen der MASA-Gruppe des IdS. Es wurde daher schon 1965 erwogen, wesentliche grammatische Daten p a r a l l e l zum Klartext auf Magnetband zu speichern und diese Daten so weit mit Informationen anzureichern, daß der entsprechende Klartext jederzeit ausgedruckt werden konnte. Diese ursprünglich sehr naiven Versuche einer Parallelcodierung - sie gehen zum Teil auf Untersuchungen zurück, die Ulrich Engel seit 1962 am Rechenzentrum der Universität Bonn durchführte - wurden unter der Anteilnahme mehrerer Mitarbeiter des IdS, unter denen Paul Wolfangel und Alex Ströbl hervorgehoben seien, vervollständigt und verbessert.

Ein relativ ausgereifter Niederschlag dieser vereinigten Bemühungen liegt vor in dem Parallelcodierungsverfahren, das Ursula Hoberg und ihre Mitarbeiter für die Untersuchungen zur deutschen Satzgliedfolge verwenden. Leistungsfähiger und vielseitiger anwendbar ist das hier von Berthold Epp vorgelegte Verfahren Parallelcodierung.

So zeigt der vorliegende Forschungsbericht den derzeitigen (fast neuesten) Stand zweier wesentlicher Bemühungen in der Abteilung Linguistische Datenverarbeitung des IdS: Der Speicherung von Klartexten und der Speicherung von künstlich codierten Paralleltexten.

Das IdS möchte mit der Herausgabe dieses Forschungsberichtes die vielen im Bereich der Linguistik tätigen Wissenschaftler auf die Verwendungsmöglichkeiten des Mannheimer Corpus aufmerksam machen. Auf der anderen Seite ist dieser Forschungsbericht auch zu verstehen als Dank an alle Mitarbeiter, die zum hier dokumentierten Stand der Arbeiten beigetragen haben.

Irmgard Vogel . Ulrich Engel

V o r w o r t

Ein weiter Weg führt vom Beginn der Erfassung unserer Texte der geschriebenen deutschen Gegenwartssprache auf Lochstreifen im Jahre 1965 bis zur Fertigstellung der vorliegenden Dokumentation.

Dazwischen liegen die 'Nachtschichten' im Deutschen Rechenzentrum in Darmstadt, auf dessen Datenverarbeitungsanlage die ersten Programme entwickelt und Ausdrucke der aufbereiteten Texte erstellt wurden. Dazwischen liegen auch die mit der Umstellung auf den IdS-eigenen Computer verbundenen Hindernisse und die mehrmalige organisatorische Neugliederung der 'Textverarbeitung' - zuerst Teil des Forschungsunternehmens 'Grundstrukturen des heutigen Deutsch', später Basis für das Demonstrationsprojekt 'Programmsystem für linguistische Aufgaben' und seit 1970 dem Großprojekt 'Linguistische Datenverarbeitung' angeschlossen.

Die in langwierigen Lernprozessen vollzogene Entwicklung der Konventionen für die Erfassung und Korrektur der Texte und die zur Einsparung von Speicherplatz und Rechenzeit erforderliche sukzessive Anpassung der wachsenden Programmbibliothek an den jeweils neuesten Stand der Programmiertechnik erforderte ein hohes Maß an Flexibilität und Geduld von allen beteiligten Mitarbeitern. Manche Erwartungen erwiesen sich als zu hoch gespannt - so z.B. der Anspruch auf hochgradige Programmportabilität - und häufig mußten Zeitpläne revidiert werden, besonders wenn es den Abschluß der Korrekturen an den Texten betraf.

Heute besteht das 'Mannheimer Korpus' der geschriebenen Gegenwartssprache aus 32 Texten mit einem Umfang von 2,2 Millionen Wörtern. Da eine ganze Reihe von Textsorten in diesem Bestand

noch nicht repräsentiert ist, wird die Datenerfassung fortgesetzt, wobei wissenschaftliche Untersuchungen zum Textsortenproblem und zur Korpusgewinnung bei der Auswahl der ergänzenden Korpustexte nach Möglichkeit berücksichtigt werden.

Seit die Texte 1970 auf die Verarbeitung mit der Rechenanlage des IdS in Mannheim umgestellt wurden, haben in jährlich zunehmendem Maße Mitarbeiter aus den verschiedenen Abteilungen und laufenden Arbeitsvorhaben des Instituts Untersuchungs- und Dokumentationsmaterial angefordert, das auf automatische Wege aus dem Textkorpus herausgezogen werden konnte. Dem institutsinternen Bedarf folgten schon bald Aufträge aus dem europäischen Ausland und den USA. Zur Bearbeitung der wachsenden Anforderungen mußte innerhalb der Abteilung Linguistische Datenverarbeitung eine Servicestelle eingerichtet werden. Als damalige Mitarbeiter dieser Servicestelle haben Klaus Bayer und Karl Kurbel die vorliegende Dokumentation zusammengestellt. Da sich auch heute noch sowohl das Textmaterial als auch die zu seiner Bearbeitung notwendigen Programme in einem Zustand stetiger Veränderung befinden, war dies keine sehr dankbare Aufgabe.

Allen Beteiligten am Aufbau der maschinell verarbeitbaren Textilbibliothek - der Datenerfassung, den Korrektoren, der Rechenzentrumsorganisation, den Programmierern, Operateuren und sonstigen Mitarbeitern, die mit Rat und Kritik die Arbeit an diesem Vorhaben unterstützten, besonders aber den Verfassern dieser Dokumentation sei für ihr persönliches Engagement und ihre Ausdauer an dieser Stelle sehr herzlich gedankt. Ohne eine gehörige Portion Uneigennützigkeit wäre das Werk wahrscheinlich heute noch nicht so weit gediehen und hätte nicht den erhofften Anklang gefunden.

Ein weiteres auf Datenträger gespeichertes Corpus vornehmli

der Zeitungssprache in Ost- und Westdeutschland, das zur Zeit ca. 1,5 Millionen umfaßt und ständig erweitert wird, hat die Bonner Forschungsstelle des Instituts für deutsche Sprache erarbeitet. Über dieses Material und darauf bezogene Forschungsergebnisse, Probleme und Projekte wird ein weiterer Forschungsbericht informieren.

Paul J. Wolfangel

Klaus Bayer
Karl Kurbel

Maschinelle Textverarbeitung im Rechenzentrum des IdS

Stand: August 1972

INHALTSVERZEICHNIS

0. VORBEMERKUNG

Bereits im Dezember 1968 berichtete I. Zint im Forschungsbericht Nr. 2 des Instituts für deutsche Sprache über maschinelle Sprachbearbeitung im Institut für deutsche Sprache (im folgenden: IdS) in Mannheim. Inzwischen - fast vier Jahre später - erscheint ein weiterer Bericht notwendig, dessen Ziel es sein soll, über die inzwischen veränderten Arbeitsbedingungen und insbesondere über die erweiterten und verfeinerten Formen der maschinellen Sprachbearbeitung zu informieren.

Dieser Bericht umfaßt *nicht* die Arbeiten zur maschinellen Textverarbeitung in der Bonner Forschungsstelle des IdS; über diese Arbeiten wird demnächst ein eigener Forschungsbericht erscheinen.

Der Bericht soll im wesentlichen dazu dienen, dem nicht in die Probleme der Programmierung eingeführten Linguisten innerhalb und außerhalb des Instituts einen Überblick über Möglichkeiten einer maschinellen Unterstützung eigener linguistischer Vorhaben durch das IdS-Rechenzentrum in Mannheim zu geben[1]. Eine solche Unterstützung kann - um Mißverständnissen vorzubeugen - nicht vorwiegend in einer Lösung theoretischer Aufgaben bestehen, sondern eher in der Bereitstellung von in geeigneter Weise aufbereitetem Sprachmaterial zur empirischen Stimulierung der Theoriebildung und zur empirischen Prüfung linguistischer Theorien.

Während 1968 noch weitgehend am Deutschen Rechenzentrum in Darmstadt gerechnet wurde, verfügt das IdS heute über eine eigene Rechenanlage vom Typ SIEMENS 4004/35. Die Textverarbeitungsprogramme sind auf die Systemsoftware dieser Anlage abgestimmt und unabhängig von den Standard-Textverarbeitungsroutinen des DRZ, die noch 1968 im Rahmen der Textverarbeitung Verwendung fanden. Die verwendeten Programmiersprachen sind: SIEMENS-4004-PBS-Assembler, -FORTRAN IV und -ALGOL 60.

Die Textbibliothek der *gesprochenen* Sprache, deren Zusammensetzung im folgenden unberücksichtigt bleiben soll (vgl. dazu die einschlägigen Veröffentlichungen der Forschungsstelle Freiburg des IdS[2]), unterscheidet sich von der Textbibliothek der geschriebenen Sprache strukturell dadurch, daß die Freiburger Texte bereits gemäß den Ergebnissen einer Voranalyse bei der Transkription der Schallaufnahmen in linguistisch relevante Einheiten segmentiert und die Segmente als Hauptsätze, abhängige Hauptsätze, Nebensätze, Parenthesen usw. klassifiziert sind. Zur Auswertung der Texte nach diesen "expliziten Kriterien" steht das Programmsystem TEK[3] zur Verfügung.

Die Texte der *geschriebenen* Sprache[4] werden dagegen nahezu ausschließlich nach der Satzzeichensetzung gemäß den Intuitionen ihrer Autoren segmentiert. Eine nähere Klassifikation der so gewonnenen Segmente ist wegen der Verschiedenheit der zu solchen Segmentierungen führenden Intuitionen nicht sinnvoll, so daß eine exaktere Auswertung der Sätze nach syntaktisch relevanten Sequenzen erst etwa nach einer Parallelcodierung (vgl. Punkt 4 der Gliederung) möglich wäre. Die Textbibliothek der geschriebenen Sprache entspricht im wesentlichen in ihrem Aufbau noch der Beschreibung von Zint, so daß die vielfach gemachten Einschränkungen bezüglich der Repräsentativität des 'Mannheimer Corpus' für die 'geschriebene Gegenwartssprache' weiterhin gelten. Die Korrekturen der Texte stehen kurz vor dem Abschluß.

Die Programme zur Durchführung der im folgenden dargestellten Textverarbeitungsschritte wurden von den Mannheimer Mitarbeitern der Projektabteilung Linguistische Datenverarbeitung erstellt. Die Textverarbeitung im Zusammenhang mit dem Mannheimer und dem Freiburger Korpus macht dabei nur einen geringen Teil der Projektarbeit aus: Zur Zielsetzung der einzelnen Arbeiten innerhalb des Projekts vgl. die Aufsätze von W.J. Backhausen[5] und G. Ungeheuer[6].

1. AUFNAHME VON TEXTEN UND ERSTE AUFBEREITUNG FÜR DIE VERARBEITUNG[7)]

Zur Eingabe in die Datenverarbeitungsanlage müssen die Texte in maschinenlesbarer Form vorliegen, d.h. auf Lochkarten, Lochstreifen oder auf Magnetbändern. Die Erfassung der Texte des Mannheimer Korpus erfolgt in der Regel auf Lochstreifen.

Lochstreifen

Die im IdS für die Texte des Mannheimer Korpus erstellten 8-Kanal-Lochstreifen (Supertyper-Code) enthalten außer dem fortlaufenden Text Informationen über die Zeilenstruktur des Originaltextes sowie über Groß- und Kleinschreibung. (Das Ende einer Zeile wird durch Wagenrücklaufzeichen, ein Großbuchstabe durch zwei Umschaltzeichen gekennzeichnet.)

Schreibkonventionen

Auf den Lochstreifen sind bereits die für die Verarbeitung von Texten der geschriebenen Sprache geltenden Schreibkonventionen berücksichtigt:

- Der Textkopf (Sigle zur Textidentifikation, Titel, Verfasser, Verlagsangaben) ist in einer bestimmten Reihenfolge angeordnet;
- Überschriften, Verfasser-, Verlags- und Quellenangaben, Datum, Zitate, Fortsetzungshinweise, fremdsprachliche Ausdrücke, drucktechnisch hervorgehobene Wörter u.a. werden gesondert gekennzeichnet;
- vor und nach einem Satzzeichen steht je ein Blank;
- jeder Satz muß mit dem Punkt (".") enden; deshalb sind teilweise Punkte zu ergänzen, z.B. nach Sätzen, die im Originaltext mit "!" oder "?" enden;
- Großschreibung am Satzanfang ist nur zulässig, wenn der Satz mit einem Nomen beginnt.

(Die hier aufgeführten Regeln stellen nur dann den wichtigsten Teil der im übrigen sehr detaillierten Konventionen dar; vgl. Anhang 5.2, S. 21 ff)

Aufnahme der Lochstreifen

Die Lochstreifeninformation wird mit dem Programm LABOL so auf Magnetband aufgenommen, daß jeder Block 100 Lochstreifenzeichen enthält.

Magnetband mit zeilennumeriertem Text

Das Programm LSCOD erstellt auf der Basis des ersten Magnetbandes (Originalband) ein neues Band im SIEMENS-EBCDI-Code, auf dem pro Block eine Zeile des Originaltextes gespeichert ist und auf dem außerdem jede Zeile des Textes eine fortlaufende Nummer erhält (LSCOD-Format). Die Wagenrücklaufzeichen und die Umschaltzeichen der Großschreibung werden eliminiert; ein Block enthält dann maximal 100 Textzeichen (einschließlich Blanks). Jede Zeile endet mit einem Blank. Das Magnetband mit zeilennumeriertem Text enthält für jede Zeile

- Information über die tatsächliche Anzahl der Zeichen in der Zeile
- die Zeilennummer
- die Textzeile selbst.

Auf dem Band sind diese Angaben neben weiteren technischen Informationen wie folgt gespeichert:

Magnetbandblock

Block-Lücke	techn. Informationen	A	B	C	Block-Lücke

A Anzahl der Zeichen in der Zeile (C)

B Zeilennummer

C Zeile, abschließend mit genau einem Blank

Weitere Datenträger

Grundsätzlich besteht die Möglichkeit, Texte von Lochkarten, Lochstreifen und Magnetbändern, auch in anderen Codes, auf Magnetband zu übertragen. Für eine Vielzahl von Datenstrukturen ist es möglich, sie in das im IdS verwendete LSCOD-Format umzusetzen. Eine Zusammenfassung der Eingabe-, Umcodierungs- und Formatierungsprogramme zu einem einheitlichen System (ALUCOS) befindet sich z.Zt. noch im Stadium der Entwicklung, wird jedoch in Kürze zur Verfügung stehen.

Gegenwärtig werden verstärkt Möglichkeiten einer Erweiterung des Korpus durch Aufnahme von *Setzlochstreifen* erwogen und erprobt, wie sie in Druckereien zum automatischen Satz Verwendung finden. Dieses Verfahren hat den Vorteil, daß die besonders arbeitsintensiven Ablocharbeiten bei der Textaufnahme entfallen. Nach der Aufnahme der Setzlochstreifen leisten verschiedene Programme eine teilautomatische Anpassung der auf den Lochstreifen vorliegenden Texte an die im IdS gültigen Schreib- und Korrekturkonventionen. Bisher wurden fünf Trivialromane und eine Samstagsausgabe des 'Mannheimer Morgen' nach diesem Verfahren aufgenommen und bearbeitet.

2. AUSDRUCK UND KORREKTUR DER TEXTE[8)]

2.1. Zeilennumerierter Ausdruck[9)]

Das Magnetband mit Zeilennumerierung wird mit dem Programm DRLINE ausgedruckt. Der zeilennumerierte Ausdruck ist eine genormte (Schreibkonventionen!) Wiedergabe des Originaltextes, bei der in der Regel eine Zeile des Ausdrucks einer Zeile des Originaltextes entspricht.

Dieser erste Ausdruck wird in aller Regel Fehler enthalten, die beim Schreiben der Lochstreifen - seltener auch durch Maschinen- oder Operateurfehler - entstanden sind. Er wird deshalb von einem Korrektor mit dem Originaltext verglichen. Der Korrektor fertigt

auf einem Formblatt ein Protokoll über die gefundenen Fehler an, aufgrund dessen Lochstreifen erstellt werden, welche die korrigierte Version der im Ausdruck fehlerhaften Zeilen einschließlich der Zeilennummern enthalten. LABOL überträgt diese Korrekturzeilen auf Magnetband, CORLS besorgt die Formatierung dieses Bandes gemäß den für zeilennumerierte Magnetbänder geltenden Konventionen. Ein vom Computerhersteller geliefertes Standard-Sortierprogramm ordnet die Korrekturzeilen nach aufsteigender Reihenfolge der Zeilennummern; das Programm CORREC erstellt ein korrigiertes Band aus den Zeilen des ursprünglichen zeilennumerierten Textes und den Korrekturzeilen.

2.2. Satznumerierter Ausdruck [10)]

Für die linguistische Auswertung ist meist eine Segmentierung des fortlaufenden Textes in überschaubare Analyseeinheiten wünschenswert. Eine der möglichen Segmentierungen ist die Segmentierung in Sätze.

Als Analyseeinheit wird in der weiteren maschinellen Verarbeitung der Texte der Satz gewählt. Die Segmentierung eines Textes in Sätze richtet sich nach der vom Autor des Originaltextes vorgenommenen Satzeinteilung: Als Satz gelten alle Sequenzen, die mit der Zeichenkombination 'Blank/Punkt/Blank' enden (vgl. oben: Schreibkonventionen). Daraus folgt, daß die Zerlegung von Texten in Sätze als Analyseeinheiten keinesfalls als linguistisch fundierte konsequente Segmentierung angesehen werden kann; die Satzzerlegung hat vielmehr nur die oben angedeutete Funktion, eine erste grobe Vorsegmentierung der Texte in überschaubare Einheiten zu liefern, die in jedem Fall unter dem Gesichtspunkt speziellerer Auswertungen einer kritischen Überprüfung bedarf. Allerdings besteht eine gewisse Wahrscheinlichkeit dafür, daß die Segmentierung in Sätze gemäß der Intuition des Originaltext-Autors in vielen Fällen mit der linguistisch-theoretisch fundierten Segmentierung übereinstimmen wird.

Zur Erstellung eines sogenannten satznumerierten Textes wird der Inhalt des Bandes mit zeilennumeriertem Text mit dem Programm SATZ in Sequenzen zwischen Punkten zerlegt und auf einem Magnetband in Blöcken variabler Länge, die jeweils einen Satz enthalten, ausgegeben. Dabei werden u.a. die speziell gekennzeichneten Zeichenfolgen für die Seitennummern des Originaltextes eliminiert; die Seitennummer erscheint nun nicht mehr zu Beginn jeder neuen Seite, sondern hinter jeder Satznummer. - Ein Magnetbandblock des Bandes mit satznumeriertem Text enthält

- Angabe über die tatsächliche Anzahl der Zeichen im Satz
- Satznummer
- Seitennummer der Seite, auf der der Satz im Originaltext beginnt
- den Satz selbst

Auf dem Band sind diese Informationen neben weiteren technischen Angaben wie folgt verteilt:

Block-Lücke	techn. Informationen	A	B	C	D	Block-Lücke

A Anzahl der Zeichen im Satz (D)
B Satznummer
C Seitennummer
D Satz

Das Programm DRSATZ druckt den Inhalt des Bandes mit satznumeriertem Text aus. Um eine den Konventionen entsprechende Zerlegung in Sätze zu gewährleisten, wird dieser Ausdruck manuell korrigiert. Die erforderlichen Korrekturen werden gemäß der oben geschilderten Verfahrensweise in den *zeilennumerierten* Text eingefügt. Ein neuer satznumerierter Ausdruck wird also erst erstellt, wenn aus dem solcher-

art korrigierten Magnetband mit zeilennumeriertem Text ein neues Satzzerlegungsband hergestellt worden ist. Dieser Ausdruck wird erneut korrigiert. - Der einzelne Text durchläuft die Verarbeitungsschleife von Ausdruck, Korrektur, erneutem Ausdruck und abermaliger Korrektur etc. so oft, bis die Fehlerquote, bezogen auf die Anzahl der Wörter im Text, unter 1 °/oo gesunken ist.

3. TEXTAUSWERTUNG

3.1. Wortformenregister [11)]

Sobald ein annähernd fehlerfreies Magnetband mit Zeilennumerierung vorliegt, läßt sich ein Wortformenregister erstellen. Die Aufbereitung des Bandes übernimmt das Programm ZEWO. Es zerlegt den in Blöcken zu je einer Zeile gespeicherten Text in einzelne Wortformen [12)] und ordnet jeder Wortform die entsprechende Zeilennummer zu. Dabei werden von vorneherein nur solche Zeichenketten berücksichtigt, die aus alphabetischen oder numerischen Zeichen bestehen.

Gewisse Ausnahmen von dieser Regel sind insofern zugelassen, als auch Folgen, die Bindestrich, Apostroph oder Abkürzungspunkt enthalten, mitübernommen werden, damit keine Wörter des Textes verlorengehen. Nicht übertragen werden in jedem Falle Satzzeichen, paarige Codierungen (z.B. die Verfasserkennzeichnung "v+ ... +v"), Doppelpunkte, die auf drucktechnische Hervorhebung hinweisen, sowie 6-stellige Codierungen mit den darauffolgenden Zeichen (z.B. die Seitenangabe ssssssOOO157).

Sofern im Rahmen einer Untersuchung nicht alle, sondern nur bestimmte Wortformen von Interesse sind, ist es möglich, ein selektives Register nur aus bestimmten gewünschten Einheiten oder ein restriktives Register aus allen bis auf bestimmte angegebene Einheiten herzustellen. ZEWO gibt ferner die Möglichkeit, die einzelnen Wortformen auf dem Zwischenband linksbündig oder rechtsbündig anzuordnen; somit

ist es möglich, neben den "normal" alphabetisch sortierten Registern auch solche zu erstellen, die rückläufig alphabetisch sortiert sind (d.h. 1. Sortierbegriff: letzter Buchstabe der Wortform, 2. Sortierbegriff: vorletzter Buchstabe etc.).

Beispiel:

Engel
fiel
viel
dunkel
Kerl
am
kam
dem
jedem

Das Standard-Sortierprogramm bringt nun die einzelnen Wortformen nach Bedarf in alphabetische oder rückläufig alphabetische Reihenfolge und gibt das Ergebnis wieder auf Magnetband aus. Das Druckprogramm AWOREG zählt die Häufigkeit des Auftretens jeder Wortform und druckt den Inhalt des sortierten Bandes in modifizierter Form aus; mehrfach vorkommende Wortformen werden nur einmal gedruckt. Das sog. Wortformenregister enthält dann bei jeder Wortform eine laufende Nummer, die Häufigkeitsangabe und die Nummern der Zeilen, in denen die Wortform im Originaltext zu finden ist.

3.2 Häufigkeitsregister [13)]

Neben der Ausgabe über den Schnelldrucker erstellt AWOREG ein Zwischenband, das gegenüber dem ursprünglichen in der Weise geändert ist, daß mehrfach auftretende Wortformen nur einmal enthalten sind und demzufolge auch keine Zeilennummern übernommen werden; zusätzlich enthält es aber bei jeder Wortform eine Häufigkeitsangabe.

Der Inhalt dieses zunächst noch alphabetisch sortierten Magnetbandes mit Häufigkeitsangaben wird nun vom Standard-Sortierprogramm nach fallenden Häufigkeiten sortiert und auf einem Ausgabeband gespeichert.

Auf dieser Basis druckt HAREG ein sogenanntes Häufigkeitsregister. Erstes Sortiermerkmal ist die Häufigkeit; bei Wortformen mit gleicher Häufigkeit wird die alphabetische Reihenfolge beibehalten.

3.3. Gemischte Register

Die bisher genannten Register bezogen sich nur auf die Verarbeitung jeweils eines Textes. Darüberhinaus ist es auch möglich, Register aus mehreren Texten bzw. aus dem ganzen Korpus herzustellen.

3.3.1. Gemischtes Wortformenregister mit Zeilennummern ("alphabetisches gemischtes Wortformenregister") [14]

Das Sortierprogramm wird in einem ersten Schritt dazu verwendet, aus den AWOREG-Eingabebändern - ein jedes enthält die Informationen, die dem Ausdruck eines Wortformenregisters zugrundeliegen - ein neues Magnetband herzustellen, das alle Wortformen der ursprünglichen Bänder nebst Quellenangabe (Sigle und Zeilennummer) enthält. Kommt also beispielsweise in Text LBT[15] die Wortform "du" 532mal vor, in Text WSA [15] 291mal etc., so sind auf dem Band folgende Informationen gespeichert:

...	du	LBT	1.Zeilennummer	du	LBT	2.ZN	du	LBT	3.ZN	...

...	du	LBT	532.ZN	du	WSA	1.ZN	...	du	WSA	291.ZN	...

AWOREG zählt nun jeweils die zu einer Text-Sigle gehörige Häufigkeit des Auftretens und druckt das gemischte Register so aus, daß mehrfach vorkommende Wortformen nur so oft gedruckt werden, wie verschiedene Siglen vorhanden sind. Neben jeder Sigle druckt AWOREG dann die Häufigkeit sowie alle Zeilennummern, unter denen die Wortform im Originaltext zu finden ist.

3.3.2. Gemischtes Register ohne Zeilennummern[16] ("gemischtes Häufigkeitsregister")

Ausgangspunkt sind die einzelnen Häufigkeitsregisterbänder, die nach fallenden Häufigkeiten sortiert sind. Diese Bänder müssen zunächst vom Sortierprogramm jeweils alphabetisch sortiert werden. Daraufhin mischt das Standard-Mischprogramm die Inhalte dieser Bänder und gibt ein gemischtes Zwischenband aus. Werden z.B.die Häufigkeitsregister des ganzen Korpus (gegenwärtig ca. 30 Texte) gemischt, so wird die Wortform "du" auf dem gemischten Band 30mal mit Häufigkeitsangaben enthalten sein. MIXREG komprimiert den Inhalt dieses Bandes so, daß "du" nur noch einmal gespeichert ist; es addiert alle Häufigkeiten, so daß neben jeder Wortform die Gesamthäufigkeit des Auftretens im ganzen Korpus enthalten ist. Das Ergebnis dieser Komprimierung gibt MIXREG wieder auf einem Magnetband aus.

Nun bestehen 2 Möglichkeiten:

1. Das Sortierprogramm sortiert das Band nach fallenden Häufigkeiten und erstellt ein sortiertes Magnetband.[17]
2. Die alphabetische Reihenfolge wird beibehalten.[18]

Der Ausdruck erfolgt (in beiden Fällen) entweder mit HAREG - pro Seite wird nur eine Spalte gedruckt - oder mit AMIREG - pro Seite werden 2 Spalten gedruckt.

3.3.3. Rückläufiges gemischtes Register

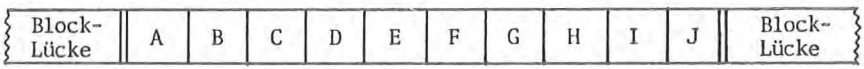
Das unter 3.1. beschriebene Verfahren, rückläufig alphabetisch sortierte Register herzustellen, läßt sich auch bei gemischten Registern anwenden. Ein Beispiel hierfür ist aus Anhang 5.13. zu ersehen.

3.4. Ausdruck von Suchbegriffen mit Kontext [19)]

Neben den bisher beschriebenen Registern, die im wesentlichen nur aus einer Auflistung einzelner Wortformen nach den genannten Kriterien bestehen, können auch solche Texteinheiten (Sätze) ausgedruckt werden, die eine gesuchte Zeichenkette (Suchbegriff) enthalten.

Mit dem Suchprogramm ASULIS wird ein Text auf vorgegebene Suchbegriffe hin abgesucht. Die gefundenen Begriffe werden mit den Sätzen, in denen sie vorkommen, ausgedruckt.

Der Text muß in konventioneller Form auf Magnetband vorliegen: Normalerweise wird von einem Magnetband mit Satznumerierung ausgegangen, das entweder nur einen Text oder nach Zusammenspielen mehrerer Magnetbänder mit dem Programm FKZ eine Reihe von Texten enthält und dann als FKZ-Band bezeichnet wird. Suchbegriffe können beliebige Zeichen oder Zeichenketten sein, also Buchstaben, Sonderzeichen, Wortformen, Satzteile, Sätze etc. Meist wird jedoch nach Wortformen abgesucht. Die Suchbegriffe werden auf Lochkarten eingelesen. Sie werden mit Kontext (Satz) auf Magnetband gespeichert. (Verwendet man als Eingabe ein Magnetband mit Zeilennumerierung, so wird als Kontext die Zeile ausgedruckt.) 1 Block enthält folgende Informationen:



A	Gesamtzahl der Zeichen des Blockes
B	Länge des Suchbegriffes (Anzahl der Zeichen)
C[20)]	Suchbegriff
D	Länge des gefundenen Wortes
E[20)]	gefundenes Wort
F	Datensatzlänge
G	Greenword
H	Länge des auszudruckenden Kontexts (Satzes)
I	Nummer des Satzes auf dem satznumerierten Magnetband
J	Kontext (Satz)

Die Ausgabe des Magnetbandinhaltes erfolgt über den Schnelldrucker, gewöhnlich im Format DIN A6. Dies hat den Vorteil, daß die Ausdrucke maschinell auf Karteikartenformat zugeschnitten werden können.

Ein Drucksegment enthält außer dem Suchbegriff und dem zugehörigen Satz eine laufende Nummer, die Satznummer sowie die Textkennzeichnung (Sigle). Zum Ausdruck wird der 112-Zeichen-Code verwendet; es besteht aber auch die Möglichkeit, im 64-Zeichen-Code auszudrucken, der gegenüber dem ersteren keine Kleinbuchstaben enthält. Zum Schluß wird eine Liste der Suchbegriffe, die Anzahl der überprüften Sätze sowie die Anzahl der gefundenen Textstellen ausgedruckt.

Während das Programm ASULIS nur unmittelbar aufeinanderfolgende Zeichenketten aufzufinden vermag, besteht darüberhinaus die Möglichkeit, nach Zeichenfolgen zu suchen, die im Text getrennt sind. Würde z.B. nach den Begriffen "heute" und "Wetter" gesucht, so würde auch ein Satz wie "heute ist das Wetter schön" ausgegeben. Das ausführende Programm trägt den Namen SUBEKO.

3.5. Schlüsselwortindex [21]

Das Programm SOREG erlaubt, eine spezielle Art von Registern zu erstellen, die - ähnlich dem KWIC-Index [22] - aus alphabetisch sortierten Wortformen im Kontext je einer Zeile bestehen.

Als Datenträger für die Eingabe findet das Magnetband mit Zeilennumerierung Verwendung, seltener wird von Lochkarten eingelesen.

Zunächst werden entsprechend dem ursprünglichen Text alle Wortformen - mit einer bestimmten Anzahl von Zeichen vor und nach der Wortform (1 Zeile) - auf einem Magnetband ausgegeben. Danach sortiert das Standard-Sortierprogramm die Zeilen in eine alphabetische Reihenfolge der Schlüsselwörter; ferner ist eine Sortierung nach den im Text enthaltenen Zeilennummern möglich.

Der Ausdruck erfolgt nach den Schlüsselwörtern zentriert auf der Basis eines solcherart sortierten Magnetbandes. Auch für diesen Index sind selektive und restriktive Ausgabemöglichkeiten vorgesehen.

3.6. Häufigkeitsstatistiken [23]

Die Häufigkeiten des Auftretens verschiedener Merkmale in einem Text lassen sich durch das Programm STAUTE ermitteln. Die Häufigkeitsstatistik kann über den gesamten Text laufen oder über ca. 1 % der Sätze im Text, die durch einen Zufallszahlengenerator anhand der Satznummern ermittelt werden.

Interessiert nur die Gesamtheit des Textes, so kann man von dem Magnetband mit Zeilennumerierung ausgehen, will man zum Vergleich eine Zufallsauswahl erhalten, so muß das Magnetband mit Satznumerierung zugrundegelegt werden. Welche Merkmale zu zählen sind, wird durch Parameter auf Lochkarte eingegeben. Merkmale können sein: Sätze, Zeilen, Wortformen, Zeichen insgesamt, Sonderzeichen, Buchstaben, Vokale, Konsonanten, Zahlen u.a.

Ferner druckt STAUTE Extrem- und Durchschnittswerte für die in einem Satz vorkommende Zahl der Wortformen und Zeichen sowie prozentuale Häufigkeiten für das Auftreten der einzelnen Buchstaben im Text aus.

4. PARALLELCODIERUNG

Die Parallelcodierung ist ein Verfahren, das es ermöglicht, einem Magnetband mit natürlichsprachlichem Text weitere Magnetbänder "parallel" zuzuordnen, so daß diese weiteren Bänder Informationen über die Ergebnisse manuell oder automatisch vorgenommener Analysen zum Originaltext enthalten[24]. Im einzelnen kann ein Text auf beliebig vielen Analyseebenen nach beliebigen Kriterien segmentiert werden (etwa in Wörter auf einer ersten Ebene, in Nominal- bzw. Verbalphrasen auf einer weiteren, auf einer dritten Ebene in Sätze), wobei den spezifizierten Textsequenzen Merkmale unter beliebigen linguistischen, psychologischen, dokumentarischen usw. Klassifikationsgesichtspunkten zugeordnet werden können.

Beispiel[25]:

Text	Parallelcodierungen		
	1. Ebene	2. Ebene	3. Ebene
Wir	PERS	NG+1P+PL+NOM	
sind	VRB+STV	VG+1P+PL+PER+ID+AK	
dann	ADV	NG+TEMP	HS
in	PREP	NG+SI+PP+LOK	
das	ART		
Brückenhaus	SUB		
gekrochen	PTZ2+STV		

Die Parallelcodierung ist als Teilnehmersystem konzipiert. Verschiedene Benutzer können das Verfahren unabhängig voneinander anwenden. Zur Vorbereitung legt jeder Benutzer einen Merkmalsvorrat zu Ebenen

fest, die er für seine Analysen als relevant ansieht. Das Beispiel zeigt, wie solche Merkmale in der Parallelcodierung formuliert werden.

Zu einem vorgegebenen Merkmalsvorrat wird automatisch ein Codeumsetzer generiert, der die Übersetzung der externen Darstellung der Merkmale in mnemonischen Kurzwörtern in die maschineninterne Darstellung in Bitketten und umgekehrt leistet. Dieser Codeumsetzer, der einem einzelnen Teilnehmer am System zugeordnet ist, kann jederzeit modifiziert werden: Neue Merkmale können aufgenommen, vorhandene gelöscht, Namen verändert werden.

Aus manuell ausgefüllten Codierformularen (vgl. zum folgenden Anhang 5.20., S. 65) werden die Merkmalsbeschreibungen mit den Bezugsdaten zum Text abgelocht. Bei der Datenaufnahme, d.h. dem Einlesen der Lochkarten, wird automatisch eine syntaktische Kontrolle der Daten (auf Ablochfehler, undefinierte Merkmale usw.) vorgenommen. Fehlerfreie Codierungen werden auf Magnetband ausgegeben; zugleich wird für alle Codierungen ein Protokoll (Anhang 5.21.2. und Seiten 119-121) ausgedruckt, wobei f o r m a l fehlerhafte Karten durch Angabe über die Art des Fehlers gekennzeichnet werden. Zur Erkennung i n h a l t l i c h e r Fehler muß dieser Ausdruck manuell bearbeitet werden.

Inhaltlich fehlerhafte Codierungen werden von dem Programm PC-40 eliminiert, die verbleibenden fehlerfreien auf einem Stammband von Codierungen gesammelt. Die bisher fehlerhaften Codierungen werden nun korrigiert und erneut abgelocht, neue Codierungen können hinzutreten. Der Verarbeitungskreislauf (siehe Anhang 5.20., S. 65) von Codieren, Lochen, Aufnahme, Korrektur und Abspeichern wird schließlich so oft durchlaufen, bis eine relativ fehlerfreie Codierung zum Text vorliegt.

Eine erste Auswertung der gesammelten Daten bereitet das Programm PC-80 vor. Es erzeugt ein strukturzeigendes Protokoll eines Textes anhand vorliegender Codierungen (siehe dazu Anhang 5.21.3., S.66), ähnlich der Darstellung im obigen Beispiel.

Weiter stehen zwei Retrieval-Programme (PC-70: Satz-orientiertes Retrieval und PC-71: Statistik-orientiertes Retrieval) zur Verfügung, die eine maschinelle Auswertung eines parallelcodierten Textes zulassen: Über bestimmte Suchbegriffe können Textsequenzen gesucht werden, zu denen Codierungen vorliegen, welche diesen Suchbegriffen genügen. Zur Formulierung der Suchbegriffe sind zwei Operatoren, nämlich "MIT" im Sinne von "muß zutreffen" für selektives Suchen und "OHNE" im Sinne von "darf nicht zutreffen" für restriktives Suchen erklärt. Die Verbindung von Merkmalen, die bis zu drei Ebenen angehören dürfen, mit diesen Operatoren beschreibt dann einen "Suchbegriff". Der Suchbegriff kann noch um die Angabe einer Wortform erweitert werden, die ebenfalls durch "MIT" bzw. "OHNE" angeschlossen wird. Eine Textsequenz genügt dann einem Suchbegriff, wenn sie alle im Suchbegriff gestellten Forderungen erfüllt. Dabei wirkt der Operator "MIT" im Sinne einer Konjunktion, der Operator "OHNE" als negierte Konjunktion. Eine Adjunktion für Merkmale aus v e r s c h i e d e n e n Ebenen ist dann durch die Ausführung mehrerer Retrievalvorgänge hintereinander, eine Adjunktion für Merkmale aus e i n e r Ebene durch Aussondern der nicht relevanten Merkmale über "OHNE" möglich.

Während PC-70 lediglich die S ä t z e liefert, in denen sich ein Suchbegriff relisiert, erlaubt PC-71 eine Steuerung der Ausgabe über einen Parameter. Es können alternativ die Sätze oder aber die Satzteil-Sequenzen ausgedruckt werden, deren Codierungen dem Suchbegriff genügen. Zugleich wird eine erste Zählstatistik durchgeführt, deren Ergebnisse wahlweise bereits für jeden zutreffenden Satz, in jedem Fall aber als Gesamtübersicht nach Beendigung des Retrieval ausgegeben werden. (Vgl. dazu Anhang 5.21.4., S. 66)

A n m e r k u n g e n

1. Für Detailinformationen, Programmbeschreibungen etc. wende man sich direkt an das Datenverarbeitungszentrum der Abteilung Linguistische Datenverarbeitung des IdS, 68 Mannheim 1, Friedrich-Karl-Straße 12

2 Steger, Hugo, Engel, Ulrich, Moser, Hugo (Herausgeber): Texte gesprochener deutscher Standardsprache I, München Hueber 1971 (Heutiges Deutsch II/1).

3 Programmnamen werden im folgenden in Großbuchstaben geschrieben.

4 vgl. Anhang 5.1., S. 20

5 Backhausen, W. J.: Linguistische Datenverarbeitung als praxisrelevante Disziplin, in: Linguistische Berichte 14 (1971).

6 Ungeheuer, Gerold: Linguistische Datenverarbeitung - die Realität und eine Konzeption, in: IBM-Nachrichten 206 (1971).

7 vgl. Anhang 5.3., S. 43

8 vgl. Anhang 5.3., S. 43

9 vgl. Anhang 5.4., S. 45

10 vgl. Anhang 5.5., S. 46

11 vgl. Anhang 5.6., S. 47 und 5.7.1., S. 48 , 5.7.2., S. 49

12 Im folgenden ist unbedingt zu beachten, daß alle Register Wort*formen*register sind. Sortiert wird nach rein graphematischen Kriterien: So werden 'Hut' in 'Er ist auf der Hut' und 'Hut' in 'Er trägt einen Hut' als identische Wortformen erkannt: Die Zugehörigkeit der Wortformen 'Hauses' und 'Häuser' zu *einer* Grundform bleibt unberücksichtigt. Eine Lemmatisierung d.h. eine Rückführung einzelner Flexionsformen auf ihre Grundform ist nicht möglich. Ebenso beziehen sich alle Häufigkeitsangaben auf Wort*formen*häufigkeiten. Die Häufigkeit, mit der ein Wort in seinen verschiedenen Flexionsformen realisiert ist, läßt sich bisher nicht automatisch ermitteln.

13 vgl. Anhang 5.6., S. 47 und 5.8., S. 50

[14] vgl. Anhang 5.9., S. 51 und 5.10., S. 52

[15] vgl. Anhang 5.1., S. 20

[16] vgl. Anhang 5.11., S. 53

[17] vgl. Anhang 5.12.1., S. 54

[18] vgl. Anhang 5.12.2., S. 55

[19] vgl. Anhang 5.14., S. 57 und 5.15., S. 58

[20] Suchbegriff und gefundenes Wort sind nicht in jedem Fall identisch. Lautet der Suchbegriff z.B. ü b e r und interessieren auch Zusammensetzungen mit dieser Graphemfolge, so werden durch entsprechende Eingabe Wörter wie ü b e r h a u p t , k o p f - ü b e r etc. bei der Suche miterfaßt.

[21] vgl. Anhang 5.16., S. 59 und 5.17., S. 60

[22] KWIC: Keyword-in-Context

[23] vgl. Anhang 5.18., S. 61 und 5.19., S. 62 ff.

[24] vgl. dazu den Entwurf zu einem Verfahren, wie dem hier skizzierten von A. Ströbl in: Forschungsberichte des IDS (Bd. 2).

[25] zu den mnemonischen Kurzwörtern vgl. Anhang 5.21.1., S. 66 ff

Anhang 5.1.

G E S C H R I E B E N E S P R A C H E

SIGLE	T E X T	ABGESCHL.	WORTANZAHL
LBT	BERGENGRUEN,DAS TEMPELCHEN	**	008663
LBC	BOELL,ANSICHTEN EINES CLOWNS	**	074132
LFH	FRISCH,HOMO FABER	**	058717
LGB	GRASS,DIE BLECHTROMMEL	**	206015
LJA	JOHNSON,DAS DRITTE BUCH UEBER ACHIM	**	087921
LMB	MANN,DIE BETROGENE	**	024276
LSO	STRITTMATTER,OLE BIENKOPP	**	111660
TJM	JUNG,DIE MAGD VOM ZELLERHOF	**	039981
TPM	PINKWART,MORD IST SCHLECHT F. H. BLUTDRUCK	**	046672
WBO	BAMM,EX OVO	**	059154
WBM	BOLLNOW,MASS UND VERMESSENHEIT DES MENSCHEN	**	070841
WGW	GAIL,WELTRAUMFAHRT	**	037339
WGS	GRZIMEK,SERENGETI DARF NICHT STERBEN	**	085312
WHK	HEIMPEL,KAPITULATION VOR DER GESCHICHTE	**	042169
WHN	HEISENBERG,DAS NATURBILD DER HEUTIGEN PHYSIK		013464
WJA	JASPERS,DIE ATOMBOMBE U. D. ZUKUNFT DES MENSCHEN	**	200580
WPE	POERTNER,DIE ERBEN ROMS	**	148079
WSP	STAIGER,GRUNDBEGRIFFE DER POETIK	**	060137
WUB	ULLRICH,WEHR DICH BUERGER	**	043524
MHE	HEUSS,ERINNERUNGEN 1905-1933	**	101776
ZFA	FRANKFURTER ALLGEMEINE ZEITUNG JAN./FEBR.1966		220112
ZWE	DIE WELT FEBRUAR 1966		062866
ZBW	BILD DER WISSENSCHAFT JAN. - MAERZ 1967	**	068005
ZSG	STUDIUM GENERALE DEZEMBER 1966	**	025920
ZUR	URANIA HEFT 11/1966 UND HEFT 1/1967		
ZB1	BILDZEITUNG JANUAR 1967	**	042331
ZB2	BILDZEITUNG FEBRUAR 1967	**	040002
ZB3	BILDZEITUNG MAERZ 1967	**	038303
ZB4	BILDZEITUNG APRIL 1967	**	038974
ZB5	BILDZEITUNG MAI 1967	**	037405
ZB6	BILDZEITUNG JUNI 1967	**	039570
ZB7	BILDZEITUNG JULI 1967	**	042804

** TEXT ABGESCHLOSSEN

Anhang 5.2.

IdS / LDV Forschungsstelle Mannheim

Schreibkonventionen und Korrekturvorschriften.
Texte der GESCHRIEBENEN SPRACHE.

A 1.1 Gestaltung des Textkopfes

A 1.1.1 Reihenfolge der Angaben und ihre Kennzeichnung

Monographie	Sammelwerk (Zeitschrift)	Zeitung
a) Zeilennumerierter Text		
ttttttSigle (3-stellig)	ttttttSigle (3-stellig)	ttttttSigle (3-stellig)
ssssssSeitenangabe	ssssssSeitenangabe	ssssssSeitenangabe
v+ Verfasser +v .	d+ Verlagsangaben +d .	d+ Verlagsangaben +d .
u+ Titel +u .	u+ Titel +u .	ssssssSeitenangabe
u+ Untertitel +u .	u+ Untertitel +u .	aaaaaaArtikelnummer
d+ Verlagsangaben +d .	ssssssSeitenangabe	u+ Titel +u .
ssssssSeitenangabe	v+ Verfasser des einzelnen Aufsatzes +v .	q+ Datum und Quellanangabe
Text	u+ Titel des einzelnen Aufsatzes +u .	ggf. Kürzel +q .
	Text	Text
b) Satznumerierter Text		
Sigle (3-stellig)	Sigle (3-stellig)	Sigle (3-stellig)
v+ Verfasser +v .	d+ Verlagsangaben +d .	d+ Verlagsangaben +d .
u+ Titel +u .	u+ Titel +u .	u+ Titel +u .
u+ Untertitel +u .	u+ Untertitel +u .	q+ Datum und Quellenangabe
d+ Verlagsangaben +d .	v+ Verfasser des einzelnen Aufsatzes +v .	ggf. Kürzel +q .
Text	u+ Titel des einzelnen Aufsatzes +u .	Text
	Text	

Bei Übergang zu neuer	d+ Verlagsangaben +d .
Nr. der gleichen Sigle:	ssssssSeitenangabe

A 1.1.2 Erläuterungen zur Gestaltung des Textkopfes

- Die Sigle des Textes muß in der ersten numerierten Zeile stehen, danach folgt kein Punkt. Vor der Sigle darf keine Leerzeile stehen.
- nach tttttt, ssssss, aaaaaa folgt die jeweilige Angabe ohne blank.
- Seitenangaben und Artikelnummern müssen aus 6 Stellen bestehen.
- nach Verfasser, Titel, Verlagsangaben, Untertitel, Quellenangaben muß jeweils ein Punkt stehen.
- zwischen jeder dieser Angaben soll eine Leerzeile stehen.
- die Verlagsangaben werden (durch Kommata getrennt) fortlaufend in folgender Reihenfolge geschrieben:

Bücher	Verlag
	Erscheinungsort
	Auflage
	Erscheinungsjahr
Sammelwerke (Zeitschriften)	Verlag
	Erscheinungsort
	Heftnummer
	Erscheinungsjahr
	Jahrgang
Zeitungen	Datum (ggf. mit Wochentag)
	Jahrgang
	Nummer
	Druck in ...

Wenn die Verlagsangaben im Original auf mehrere Seiten verteilt sind, werden sie in der Abschrift zusammengefaßt. Als Seitenangabe ist die 1. Seite der Verlagsangabe zu nehmen.

A 1.2 Zur Gestaltung der Textseiten

Seitenanfang	Leerzeile ssssss.....(6-stellige Zahl) Leerzeile Text
Absatz / Abschnitt	Ein neuer Abschnitt wird durch Einrückung der 1. Zeile um 6 Stellen gekennzeichnet. Keine Leerzeilen! (einzige zulässige Einrückung)
Tabellen (siehe auch bei A 1.5 Kennzeichnung)	Tabellen stehen zwischen t+.....+t . sie sind zeilenweise zu schreiben, d.h. für jede Tabellenzeile wird beim Schreiben eine neue Zeile begonnen. Jede Tabellenzeile wird mit einem Punkt abgeschlossen. Zwischenräume sind bis auf je eine Leerstelle vor und hinter den einzelnen Angaben durch Bindestriche aufzufüllen.

Beispiel:

	Luftfeuchtigkeit	Temperatur
Bandung	90	24
Djakarta	85	29
Bandjermasin	95	29

ist wie folgt zu schreiben:

```
t+ ------------- Luftfeuchtigkeit --- Temperatur .
Bandung --------- 90 ----------------- 24 ------- .
Djakarta -------- 85 ----------------- 29 ------- .
Bandjermasin ---- 95 ----------------- 29 ------- +t .
```

Briefe	Absender, Datum und Adresse werden fortlaufend geschrieben und zwischen q+ und +q . eingeschlossen. Anrede, Schlußformel und Unterschrift werden wie im Original geschrieben. Am Ende des Briefes wird ein Punkt gesetzt.
Beischriften zu Bildern entfallen. (siehe auch bei A 1.5 Kennzeichnung)	Beischriften zu im Text vorkommenden Abbildungen werden, wenn sie zwischen zwei abgeschlossenen Sätzen auftreten, unmittelbar an der auftretenden Stelle, wenn sie zwischen einem unvollendeten Satz auftreten erst am Satzende mit c+ Abb. +c . gekennzeichnet.
Zeilenende	Wenn bei genauer Einhaltung der auf 100 Zeichen (incl. Leerstellen) festgelegten Zeilenlänge nur Satzzeichen abgetrennt würden, ist das letzte Wort mit in die neue Zeile zu schreiben.
Zeilenende und Seitenende	Keine Trennung von Einzelwörtern. Das im Original getrennte Wort kommt in die alte Zeile, bei Überschreiten der festgelegten Zeilenlänge in die neue Zeile. Durch Bindestrich getrennte Doppelwörter können auch im Ausdruck getrennt werden.
Fortsetzungshinweise bei Zeitungen u. Zeitschriften	werden als eigenständige Sätze behandelt, sie stehen zwischen c+ und +c und werden mit Punkt abgeschlossen.
Beispiel:	c+ Fortsetzung Seite 6 +c . c+ Fortsetzung von Seite 1 +c .

Es ist verfahrensmäßig zu unterscheiden zwischen abgeschlossenen und unterbrochenen Sätzen.
Ist der Satz vor dem Hinweis abgeschlossen, so folgt der Fortsetzungshinweis unmittelbar. Bei unterbrochenen Sätzen wird der Rest des Satzes von der Fortsetzungsseite vorgezogen. Dann folgt der Fortsetzungshinweis.

A 1.3 Satzzeichen

A 1.3.1 Liste der Satzzeichen:

.

,

;

: außer als Kennzeichnung von drucktechnischer Hervorhebung!

?

!

- Gedankenstrich

"

Anführungsstriche "

'

(

)

wird ersetzt durch Doppel-Rundklammer ((

/

... Auslassungspunkte

A 1.3.2 Behandlung von Satzzeichen

Zwischen Satzzeichen und Wort (Zahl) bzw. Satzzeichen und Satzzeichen steht immer ein blank.
Das letzte Zeichen eines Satzes ist immer Punkt.
Deshalb sind in der Abschrift am E n d e d e s S a t z e s ggf. folgende Umstellungen bzw. Punkt-Ergänzungen erforderlich:

Original	Abschrift	
."	" .	
.)	) .	Punkt wird umgesetzt
.-	- .	
!"	! " .	
?"	? " .	
-	- .	Punkt wird ergänzt
)	) .	
...	... .	

Auch Buchtitel, Untertitel und Überschriften gelten als Satz und sind mit Punkt zu schließen.

Ausnahme: Überschriften, die *Teil des 1. Satzes* eines Textes sind.

Beispiel:

Original	Abschrift
WIR SIND DIE BESTEN sagte Uwe gestern zu unserem Reporter	u+ wir sind die Besten +u sagte Uwe gestern zu unserem Reporter .

A 1.3.3 Regelung bei Aufzählungen

Grundsätzlich gelten Aufzählungsmerkmale als eigene Sätze und bekommen einen Satzpunkt.

Original	Abschrift
1.	1. .
2.	2. .
3.	3. .

Aufzählung

Vorbemerkung

Im folgenden gelten als Aufzählungsmerkmale Ziffern und Buchstaben und andere abdruckbare Zeichen, wie Gedankenstriche

z.B.	1.	oder	A)	oder	a)	oder	-
	2.		B)		b)		-
	3.		C)		c)		-

Unabdruckbare Zeichen, wie fettgedruckter Punkt, Dreiecke, Balken usw. sind zu vernachlässigen.

Als Einleitung zu einer Aufzählung wird der Satzteil oder die Wortgruppe bezeichnet, die der Aufzählung vorangeht.

z.B.	Daher folgende Regelung:
oder	Also folgt:
oder	Wir kommen zu folgendem Ergebnis:

1. Aufzählung ohne Einleitung

Grundsätzlich gelten Aufzählungsmerkmale als eigene Sätze und bekommen einen Satzpunkt.

Original	Abschrift
1.	1. .
2.	2. .
3.	3. .

Wenn die Aufzählungsmerkmale jedoch wie im folgenden Beispiel in den Satzzusammenhang eingehen, wird kein zusätzlicher Satzpunkt gesetzt.

Beispiel (Original = Abschrift)

1. habe ich es nicht gesagt,
2. würde ich so etwas nie sagen,
3. bitte ich um Entschuldigung, wenn ich es gesagt habe.

2. Aufzählung mit Einleitung

Um in der Aufzählung selbst grammatikalische Sätze zu erhalten und nicht durch Einbeziehung der Einleitung in den folgenden Satz in diesem eine falsche Satzgliedstellung zu bekommen, wird die Einleitung vor der Aufzählung durch Satzpunkt abgeschlossen, wenn die Aufzählung einen oder mehrere eigenständige Sätze darstellt.

Beispiel 1

Daher folgende Regelung : .
1. . Wir werden die Texte möglichst bald abschließen .
2. . Von den folgenden Texten werden H.R. gedruckt .
3. . Die Texte werden , wenn gewünscht , gebunden .

Wenn bei Aufzählungen mit Einleitung die Aufzählungsmerkmale in den Satzzusammenhang eingehen, gilt die o.a. Regelung. (kein zusätzlicher Satzpunkt für Aufzählungsmerkmale)

Beispiel 2 (Original = Abschrift)

Daher folgende Regelung : .
1. werden wir die Texte möglichst bald abschließen .
2. werden von den folgenden Texten H.R. gedruckt .
3. werden die Texte , wenn gewünscht , gebunden .

Abschnittsgliederungsmerkmale, die ohne Einleitung größere Passagen (z.B. mehrere Seiten) einleiten und daher nicht eigentlich als Aufzählung betrachtet werden können, werden (wenn sie nicht in den Satzzusammenhang eingehen) wie Aufzählungsmerkmale behandelt.

Beispiel:	Original	Abschrift
	1)	1) .
	2)	2) .
	a)	a) .
	b)	b) .

A 1.4 Zeichen mit besonderer Regelung

A 1.4.1 Drucktechnische Hervorhebung

wird durch Doppelpunkt ohne blank am Ende des hervorgehobenen Wortes gekennzeichnet.
Als Hervorhebung gilt jede Abweichung vom Druckbild des übrigen Textes durch

a.) andere Schrifttypen wie z.B. Kursiv
b.) andere Schriftstärke wie halbfett, fett u.a.
c.) Unterstreichung, Spatiierung (Sperrung) oder Großschreibung ganzer Wörter. In Majuskeln geschriebene Wörter werden normal geschrieben.

Beispiel:	Original	Abschrift:
	BILD	Bild:

d.) Majuskeln in Abkürzungen
werden wie folgt geschrieben:

Original	Abschrift
MdB	MdB:
SPD	SPD:
GmbH	GmbH:

Drucktechnisch hervorgehobene Elemente von Komposita mit und ohne Bindestrich werden wie folgt geschrieben:

Original	Abschrift:
SPIEGEL-Redaktion	Spiegel-Redaktion:
ABhören	Abhören:

A 1.4.2 Bindestriche, Abkürzungspunkte, Ordnungszahlpunkte, Apostrophe
haben kein vorangehendes blank.

Beispiele:

1.
wie geht's

Schrägstriche werden ohne blanks geschrieben, wenn eine enge Bindung der Elemente unmittelbar vor und hinter dem Schrägstrich besteht.

Beispiele:

CDU/CSU 1966/67

Aber:

Hamburg / München 28. Februar / 1. März

A 1.4.3 Zusammengesetzte Substantivierungen
werden entsprechend dem Original geschrieben.

Beispiele:

das Mit-sich-geschehen-Lassen
ein Zu-wenig
ein Für-richtig-Halten
die Selbst-werdung

A 1.4.4 Worttrennung
Bei Trennung durch "und", "oder", Anführungsstriche, o.ä. sind ggf. Bindestriche wie in den Beispielen zu ergänzen.

Beispiel:

Original	Abschrift
Wald- und Wiesen-Tee	Wald- und Wiesen-Tee
"Entwicklungs" Völker	" Entwicklungs- " -Völker
(Wasser)ball	(Wasser-) -Ball
clearing-Stelle	f+ clearing- +f -Stelle

Wenn zu ergänzende Buchstaben einer Abkürzung durch Klammern abgetrennt sind, wird kein blank geschrieben.
z.B.: f(iliae)

A 1.4.5 Zahlen und Zahlenverhältnisse
Dezimalzahlen gelten als ein Wort.
(keine blanks, etwa vor und nach Komma)

Beispiel: 123,45

Amerikanische Zahlen

Der angelsächsische Dezimalpunkt wird als Dezimalkomma geschrieben.

Beispiel:	Original	Abschrift
	4.2	4,2

Im Text durch Punkt oder blank gegliederte Zahlen werden entsprechend den Beispielen geschrieben.

Beispiel:	Original	Abschrift
	150 000	150000
	1.250.000,25	1.250.000,25

gemischte Zahlen erhalten einen Bindestrich

Beispiel:	Original	Abschrift
	$1\frac{1}{2}$	1-1/2

Zahlenverhältnisse und Zeiten (Sportergebnisse)

Beispiel:	Original	Abschrift
	2:3	2-/-3
	der 0:1 Sieg	der 0-/-1 Sieg
	4:12,6 min	4.12,6 min
	4:13:6 Std.	4.13:6 Std.

A 1.4.6 Mathematische Ausdrücke

Mathematische Operatoren werden ausgeschrieben.

Beispiel:	Original	Abschrift
	+	plus
	-	minus
	.	mal
	:	durch
	=	gleich

Original	Abschrift
4.2-6:3+1=7	4 mal 2 minus 6 durch 3 plus 1 gleich 7
10^5	10E5

Römische Zahlen

werden als solche gelocht, also

I, IV, X usw. (nicht 1, 4, 10).

Vor jeder römischen Zahl steht ohne blank das Zeichen @

A 1.4.7 Formeln

Original	Abschrift
N_2H_4	N2H4
$ZN(NH_3)_2CL_2$	((ZN(NH3)2CL2))

Eckige Klammern siehe unter nicht abdruckbare Zeichen (A 1.5.5)

A 1.5 K e n n z e i c h n u n g

Zur Kennzeichnung werden verwendet:

```
c+ ........ +c
f+ ........ +f
q+ ........ +q
t+ ........ +t
u+ ........ +u
v+ ........ +v
x+ ........ +x
z+ ........ +z
```

Erstreckt sich eine zu kennzeichnende Textpassage über mehrere Sätze, so genügt eine Kennzeichnung am Anfang und Schluß der Passage.
Aus früheren Regelungen herrührende satzweise Kennzeichnungen brauchen nicht geändert zu werden.

A 1.5.1 Fremdsprache, Dialekt und sonstige nicht-hochsprachliche Wörter, Sequenzen oder Sätze.

Zeichen: f+ ..... +f

Alle Wörter, die nicht im Duden oder Fremdwortduden stehen, gelten als fremdsprachlich.

Das gilt auch für Dialektwörter.

Als nicht-hochsprachlich gelten Abweichungen von normalen grammatischen Formen wie z.B.:

Original	Abschrift
Haste?	f+ Haste +f ?
Kommste?	f+ Kommste +f ?
nu	f+ nu +f

Namen und Bezeichnungen außerhalb fremdsprachlicher Ausdrücke werden vorläufig nicht besonders gekennzeichnet.

Es werden gekennzeichnet:

1. einzelne Wörter: wenn nur ein einzelnes Wort oder eine relativ geringe Zahl von Wörtern a) eine Abweichung von grammatischen Formen aufweist oder eindeutig ein Dialektwort b) fremdsprachlich ist;

2. der ganze Satz: wenn der überwiegende Teil des Satzes Dialekt oder fremdsprachlich ist.

Beispiele für 1. Und so schön schreiben f+ kanner +f . Mann , brauchen f+ Se +f doch f+ nich +f rot zu werden , wenn ich Feuer haben will .

Ziemlich alles , was f+ portable +f ist , findet sich in den f+ trailers +f die ja selbst im Grunde f+ portable homes +f sind .

Beispiele für 2.

f+ Und middem Daumen jeh ich rein, ganz automatisch und kleb Nägel und Knöppe zwischen, und vor dreiunddreißig hatt' ich ne Zeit , da habe ich Stacheldraht auf Zinnober jesetzt +f .

f+ Han ich dich nich jesacht , dat issen Jimmy +f ! .

f+ it is a pleasure to acknowledge my indebtedness to those who have contributed in various ways to the preparation of this paper +f .

A 1.5.2 Zitate

Zeichen z+ ..... +z

Als Zitate gelten (nach Duden) wörtliche Anführungen von Textstellen aus einem Buch, Schriftstück, Brief u. ä., z. B. Gedichte.

Darunter fallen also nicht:

- bloße wörtliche Rede des lfd. Textes
- durch Anführungsstriche gekennzeichnete Titel, Namen und termini technici
- durch Anführungsstriche lediglich gekennzeichnete ungewöhnlich verwendete Wörter und Fügungen (abgeschwächte Rede, ironische Redeweise, Ungefährformulierungen usw.)

Beispiele für termini technici, die im fortlaufenden Text durch Anführungsstriche gekennzeichnet werden:

> In der Industrie ist das "Systemdenken" auch schon dort eingedrungen, wo das Konzept vom gesamten System, d. h. der gesamten Kooperation schwierig zu umreißen ist.
> Wir sind jetzt an dem Punkt angelangt, wo wir, wie so viele Leute, das System selbst als einen "schwarzen Kasten" ansehen können

> Das Versagen des 'Minimierungsprinzips" ist auf das Versagen zurückzuführen, die wahren Probleme zu konkretisieren, die mit den Fragen des Alkoholismus nun einmal verbunden sind.

Das Anführungszeichen wird hier verwendet, um dem Leser die Information zu geben, daß der betreffende Ausdruck in dem Begriffsinhalt des genormten Fachausdrucks zu verstehen ist.

A 1.5.3 Datum von Quellenangaben

Zeichen: q+ ..... +q

Beispiel: q+ Bonn , den 28. 1. 1964 (dpa) +q .

A 1.5.4 Verfasserangaben in Zeitungsartikeln (ausgeschrieben, abgekürzt oder als Siglen)

Zeichen: v+ ..... +v

Beispiel: v+ -Ke +v .

Die Verfasserangabe ist in der Abschrift zwischen Überschrift und Textanfang vorzuziehen, auch wenn sie im Original am Ende des Artikels steht.
Innerhalb einer Quellenangabe braucht der Verfasser nicht besonders gekennzeichnet zu werden.

A 1.5.5 Nichtabdruckbare Symbolzeichen

wie z.B. ^{o} (Grad), (Pfund), § (Paragraph) werden ausgeschrieben; das ausgeschriebene Symbol wird zwischen x+ und +x gesetzt.

Beispiel:	Original	Abschrift
	30^{o}	30 x+ Grad +x

Eckklammern werden durch je 2 Rundklammern ersetzt.

Beispiel:	Original	Abschrift
	Morphem	((Morphem))

A 1.5.6 Häufung von Kennzeichnungen

Häufen sich an einer Textstelle mehrere Kennzeichnungen, die Wörter, Satzteile oder Sätze einschließen, so besteht in der Reihenfolge grundsätzlich kein Vorrang bestimmter Zeichen.
Bei fremdsprachlichem Zitat in Fragesatzform ist z.B. folgende Kennzeichnung möglich:

z+ f+ ..... ? +f +z
f+ z+ ..... ? +z +f

Es ist natürlich darauf zu achten, daß die Zeichen untereinander symmetrisch angeordnet werden:

u+ Neues aus f+ far-west +f +u .

A 2 ZUSÄTZLICHE HINWEISE FÜR DIE KORREKTUR

A 2.1 *Offensichtliche Druckfehler* sind zu korrigieren, nicht jedoch *grammatische Fehler*
wie Verwechslung von Singular und Plural, falsche Rektion etc.

z.B. Wehner und Schmidt fährt nach Moskau.
Ein solcher grammatischer Fehler darf nicht geändert werden.

A 2.2 Die Fehler sind beim Korrigieren im Text leserlich (z.B. dünner roter Filzstift) durch Unterstreichen der betr. Stelle und der laufenden Zeilennummer oder Satznummer am Rande zu kennzeichnen.

A 2.3 Die Fehler sind (mit kurzer Beschreibung der Art des Fehlers) auf ein Korrekturprotokoll zu notieren.

A 2.4	Textende bei zeilennumeriertem Ausdruck	ttttt-Korrekturstand-Datum-Name des Korrektors z.B. ttttt-dritte-Korrektur-15.08.71-Maier

A 2.5 Die Korrektur muß immer nach dem Originaltext vorgenommen werden. Es muß darauf geachtet werden, daß die richtige Ausgabe und Auflage verwendet wird (Verlagsangabe!)

A 3 SCHREIBTECHNISCHE ANWEISUNGEN

A 3.1 Am Anfang und Ende eines jeden Streifens muß etwa je eineinhalb Meter ungelocht bleiben; mindestens je ein Meter davon muß Führungslöcher enthalten.

A 3.2 Abgerissene Streifen dürfen nicht mit Klebstoff, sondern nur mit den eigens dafür vorgesehenen Klebestücken geflickt werden.

A 3.3 Nach dem Umschalten darf der nächste Buchstabe nicht zu schnell angeschlagen werden.

A 3.4 Der Wagenrücklauf darf nie von Hand betätigt werden, er erscheint sonst nicht auf dem Streifen. Vorsicht beim Neueinschalten der Maschine (morgens)!

A 3.5 Insgesamt muß *staccato* geschrieben werden, d.h. die einzelnen Anschläge müssen deutlich zeitlich getrennt sein. Lieber zu langsam als zu schnell schreiben! Zu rasche Aufeinanderfolge von Anschlägen ergibt häufig Fehllochungen.

A 4 ERSTELLUNG VON KORREKTURSTREIFEN

A 4.1 Es wird zeilenweise korrigiert.

A 4.2 Jede im Korrekturausdruck angestrichene Zeile muß neu geschrieben werden. Zwischen laufender Nummer und Text müssen 2 Leerstellen sein. Dadurch wird automatisch die falsche Zeile des Ausdrucks gelöscht und die verbesserte Zeile eingefügt.

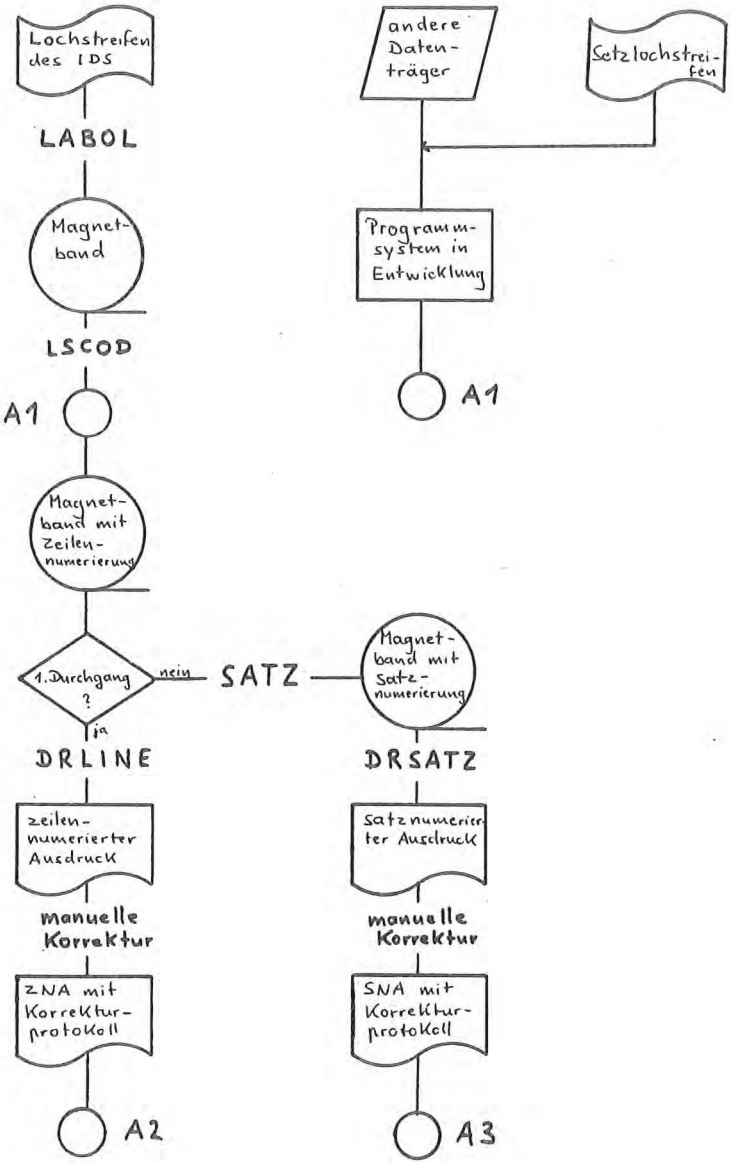
A 4.3 Muß eine Zeile des Ausdrucks *gelöscht* werden, so braucht nur die laufende Zeilennummer mit 2 folgenden blanks eingetippt zu werden.

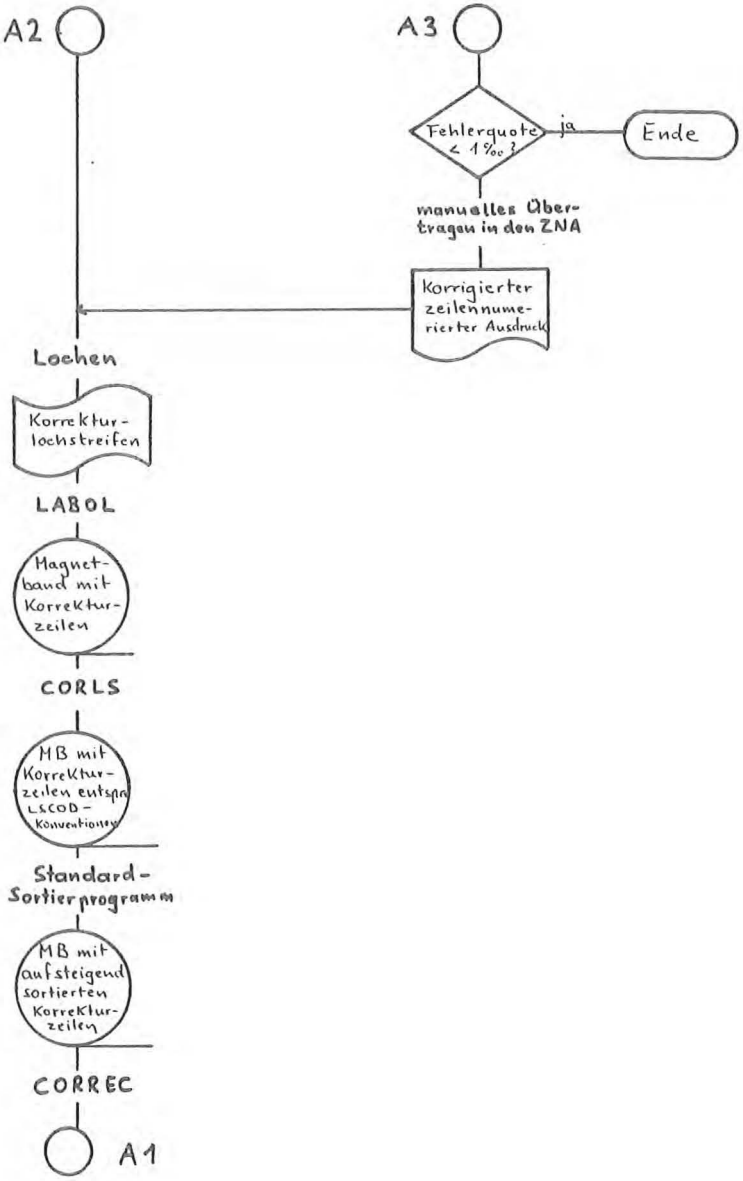
A 4.4 Hat man auf dem Korrekturstreifen einen Fehler gemacht, so schreibt man die laufende Nummer und den richtigen Text neu. Die Maschine berücksichtigt, wenn auf dem Korrekturstreifen mehrere Zeilen mit derselben Nummer stehen, immer nur die letzte.
Ist auf dem Korrekturstreifen nur die lfd. Nummer falsch geschrieben, so wird diese falsche Nummer neu geschrieben und ohne Leerstellen 6 blank angehängt. Dabei ist zu beachten, daß alle eventuell vorhergehenden Zeilen mit der falschen Nummer ebenfalls gelöscht werden. Notfalls müssen also auch diese Zeilen neu geschrieben werden.

A 4.5 Die Umschalttaste ist, wenn die laufende Nummer geschrieben ist, in jedem Falle loszulassen (und bei Beginn des Textes gegebenenfalls neu zu drücken).

Anhang 5.3.: TEXTAUFNAHME, AUSDRUCK UND KORREKTUR

vereinfachter Datenflußplan




A2
A3
Fehlerquote < 1‰ ?
ja
Ende
manuelles Übertragen in den ZNA
Korrigierter zeilennumerierter Ausdruck
Lochen
Korrektur-lochstreifen
LABOL
Magnet-band mit Korrektur-zeilen
CORLS
MB mit Korrektur-zeilen entspr. LSCOD-Konvention
Standard-Sortierprogramm
MB mit aufsteigend sortierten Korrektur-zeilen
CORREC
A1

Anhang 5.4.: ZEILENNUMERIERTER AUSDRUCK

AZENTE V001 AUSDRUCK NACH DER KORREKTUR DATUM 12.11.73 SEITE 0006

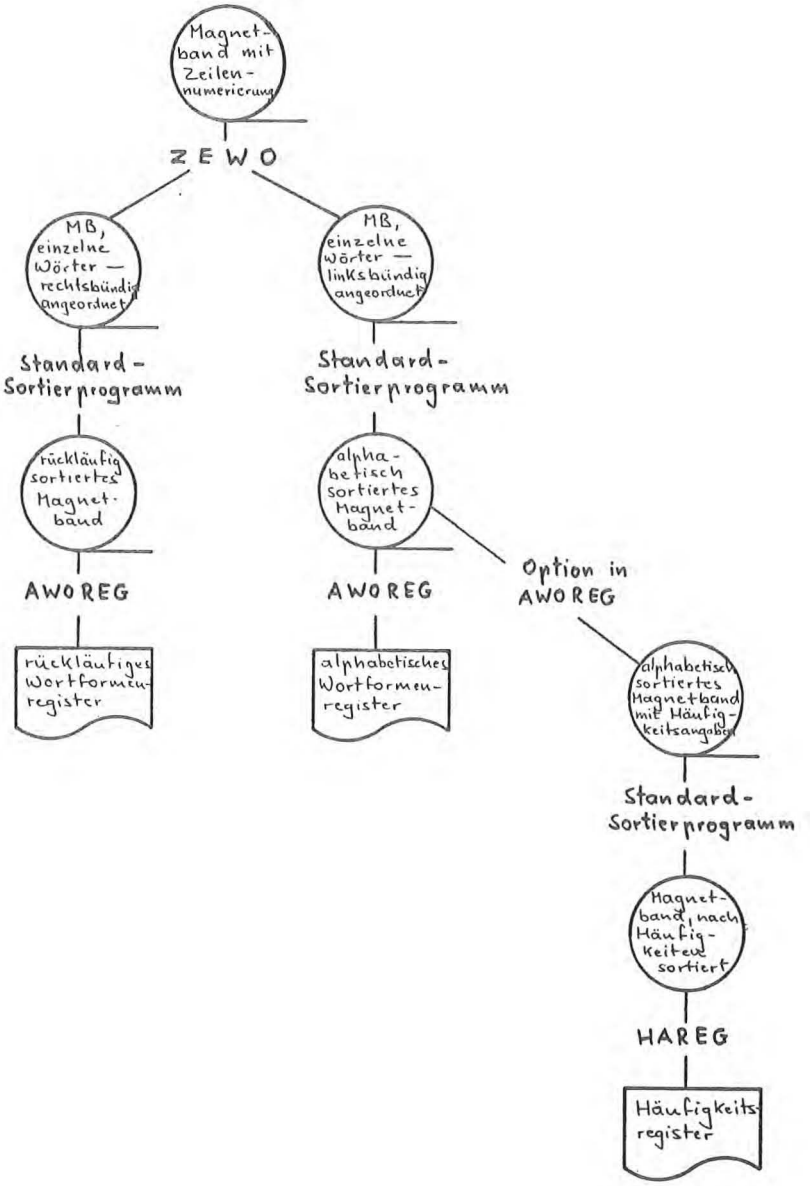
```
001370  mit ihnen ausgekommen . Großpapa hatte so militärische
001380  Ansichten , weißt du , der Kaiser hier und
001390  der Kaiser dort , und niemand soll räsonnieren . nun ,
001400  damals ist das so gewesen , heute denken viele anders ,
001410  und manche auch schon allzusehr . ich meine immer ,
001420  auf das Herz des Menschen kommt es an , und nicht
001430  auf seine Sprache oder auf die Gedanken in seinem
001440  Kopf . die Gedanken in seinem Kopf , die können
001450
001460  ssssss000012
001470
001480  falsch sein , das ist wie mit der Orthographie oder
001490  dem höheren Rechnen , wo man sich auch sehr täuschen
001500  kann . aber in seinem Herzen hat der Mensch
001510  einen Punkt , da kann er nicht irren . und an dem
001520  kann man ihn erkennen .
001530        damals gab sich Großpapa schon viel Mühe um
001540  mich , und natürlich gefiel mir das . aber ich sah ihn
001550  doch nicht sehr häufig , weil er immer bei der Brücke
001560  war . ganz versessen war er auf diesen Bau ! .
001570        der Sommer ging schon in den Herbst über , es
001580  war naß und kühl . eines Morgens war ich ganz
001590  früh im Park herumgegangen  und dann im Walde
001600  und dann wieder im Park . ich hatte gemerkt , daß
001610  Großpapa richtige Absichten auf mich hatte , und
001620  auch meine Eltern hatten mir so etwas angedeutet ,
001630  und da meinte ich , ich müßte in der Einsamkeit und
001640  in der Natur mit mir zu Rate gehen , wahrscheinlich
```

ASANTE V001 AUSDRUCK SATZZERLEGUNG DATUM 12.11.73 SEITE 0003

SATZ	SEITE	ZEILE	
		2	oder drei polnische Gutsbesitzer , die Bauern waren meistens Rechtgläubige , und katholische Leute
		3	fand man selten ; wie das jetzt ist , das wirst du besser wissen als ich .
39	000011	1	Großpapa liebte die Polen nicht und sagte , sie seien Aufrührer von Natur , aber ich bin immer gut
		2	mit ihnen ausgekommen .
40	000011	1	Großpapa hatte so militärische Ansichten , weißt du , der Kaiser hier und der Kaiser dort , und
		2	niemand soll räsonnieren .
41	000011	1	nun , damals ist das so gewesen , heute denken viele anders , und manche auch schon allzusehr .
42	000011	1	ich meine immer , auf das Herz des Menschen kommt es an , und nicht auf seine Sprache oder auf die
		2	Gedanken in seinem Kopf .
43	000011	1	die Gedanken in seinem Kopf , die können falsch sein , das ist wie mit der Orthographie oder dem
		2	höheren Rechnen , wo man sich auch sehr täuschen kann .
44	000012	1	aber in seinem Herzen hat der Mensch einen Punkt , da kann er nicht irren .
45	000012	1	und an dem kann man ihn erkennen .
46	000012	1	damals gab sich Großpapa schon viel Mühe um mich , und natürlich gefiel mir das .
47	000012	1	aber ich sah ihn doch nicht sehr häufig , weil er immer bei der Brücke war .
48	000012	1	ganz versessen war er auf diesen Bau ! .
49	000012	1	der Sommer ging schon in den Herbst über , es war naß und kühl .
50	000012	1	eines Morgens war ich ganz früh im Park herumgegangen und dann im Walde und dann wieder im Park .
51	000012	1	ich hatte gemerkt , daß Großpapa richtige Absichter auf mich hatte , und auch meine Eltern hatten mir
		2	so etwas angedeutet , und da meinte ich , ich müßte in der Einsamkeit und in der Natur mit mir zu
		3	Rate gehen , wahrscheinlich hatte ich irgendwo gelesen , daß man das so macht .
52	000012	1	ich war noch sehr jung , jünger als heutzutage die Mädchen zu sein pflegen , um die man anhält .
53	000012	1	du warst , glaube ich , zwanzig , nicht wahr ? .
54	000012	1	mir scheint , ich bin an diesem Morgen nicht sehr weit gekommen mit meinen Überlegungen , ich war
		2	auch noch so kindisch , daß ich mich von jedem Eichhörnchen ablenken ließ .
55	000012	1	dann fand ich es an der Zeit , zum Frühstück ins Haus zu laufen , aber da ging plötzlich ein Schauer
		2	nieder , ich war gerade in der Nähe des Tempelchens , und so wollte ich dort Schutz suchen .
56	000013	1	du sagst ja , daß du ein Bild von ihm gesehen hast .
57	000013	1	da weißt du , daß es im Gebüsch versteckt lag und gerade so groß war , daß eine kleine Gesellschaft
		2	dort Tee trinken konnte ; der Pavillon ist stattlicher gewesen , aber damals ging alles noch
		3	bescheidener zu .
58	000013	1	wir benutzten das Tempelchen selten , weil meine Mutter fand , es herrsche dort ein fauliger Geruch .
59	000013	1	vielleicht war auch der Schwamm darin .

Anhang 5.6.: WORTFORMEN- UND HÄUFIGKEITSREGISTER

vereinfachter Datenflußplan



Anhang 5.7.1.: Alphabetisch sortiertes Wortformenregister

ALPHABETISCHES REGISTER MIT HAEUFIGKEITSANGABE BERGENGRUEN, DAS TEMPELCHEN SEITE 18
12:11:73

LAUF-NR.	WORTFORM:	HAEUFIGKEIT:	ZEILENINDEX:							
000285	drei	* 7 *	000710	001310	005710	005960	008480	009960	010020	
000286	dreist	* 1 *	005880							
000287	dringen	* 1 *	008240							
000288	drolliger	* 1 *	006580							
000289	du	* 23 *	000610	000690	001180	001330	001380	001670	001800	001800
		*	001810	002520	006660	007500	009550	009750	009970	010190
		*	010310	010390	010640	010660	010670	011080	011720	
000290	dumm	* 1 *	007830							
000291	dumme	* 1 *	006710							
000292	dummen	* 1 *	010940							
000293	dunkel	* 1 *	004870							
000294	dunkelbraun	* 1 *	002410							
000295	durch	* 8 *	000150	000300	002380	002920	007500	007560	007900	007950
000296	durchgekommen	* 1 *	007270							
000297	durchsucht	* 1 *	004300							
000298	durfte	* 1 *	006680							
000299	eben	* 4 *	001290	005450	010290	011300				
000300	ebenso	* 1 *	010470							
000301	ebensowenig	* 1 *	004940							
000302	ehe	* 1 *	008870							
000303	ehemaligen	* 1 *	010250							
000304	eher	* 1 *	008360							
000305	ehrenhafter	* 1 *	009110							
000306	eigene	* 1 *	000500							
000307	eigenen	* 2 *	010980	011300						

Anhang 5.7.2: Rücklaufig alphabetisch sortiertes Wortformen-

RUECKLAEUFIGES REGISTER MIT HAEUFIGKEITSANGABE LBT,DAS TEMPELCHEN SEITE 26
12:11:73

LAUF-NR.	WORTFORM:		HAEUFIGKEIT:		ZEILENINDEX:							
000471	Kopf	*	5	*	001440	001440	002020	011150	011500			
000472	darf	*	3	*	002010	005600	005650					
000473	auf	*	38	*	000370	000770	000890	000930	001420	001430	001430	001560
				*	001610	001920	001950	002010	002600	003950	004520	004810
				*	005260	005420	006500	007170	007230	007610	008000	008010
				*	008060	008590	008890	009360	009390	009740	010150	010270
				*	010700	010910	011580	011690	011780	011990		
000474	hinauf	*	1	*	002480							
000475	darauf	*	2	*	003730	008950						
000476	Beruf	*	1	*	012090							
000477	Vogelruf	*	1	*	007910							
000478	lag	*	10	*	000640	001910	001930	001950	003510	003950	005420	006110
				*	008770	011240						
000479	Verlag	*	1	*	000061							
000480	mag	*	1	*	008250							
000481	Nachmittag	*	1	*	008050							
000482	Tag	*	3	*	000990	004610	004840					
000483	Weg	*	3	*	007490	008090	008160					
000484	ungläubig	*	1	*	009890							
000485	Zweig	*	1	*	009910							
000486	häufig	*	1	*	001550							
000487	geläufig	*	1	*	003140							
000488	geradwegig	*	1	*	004020							
000489	völlig	*	1	*	011090							
000490	unwillig	*	1	*	002170							

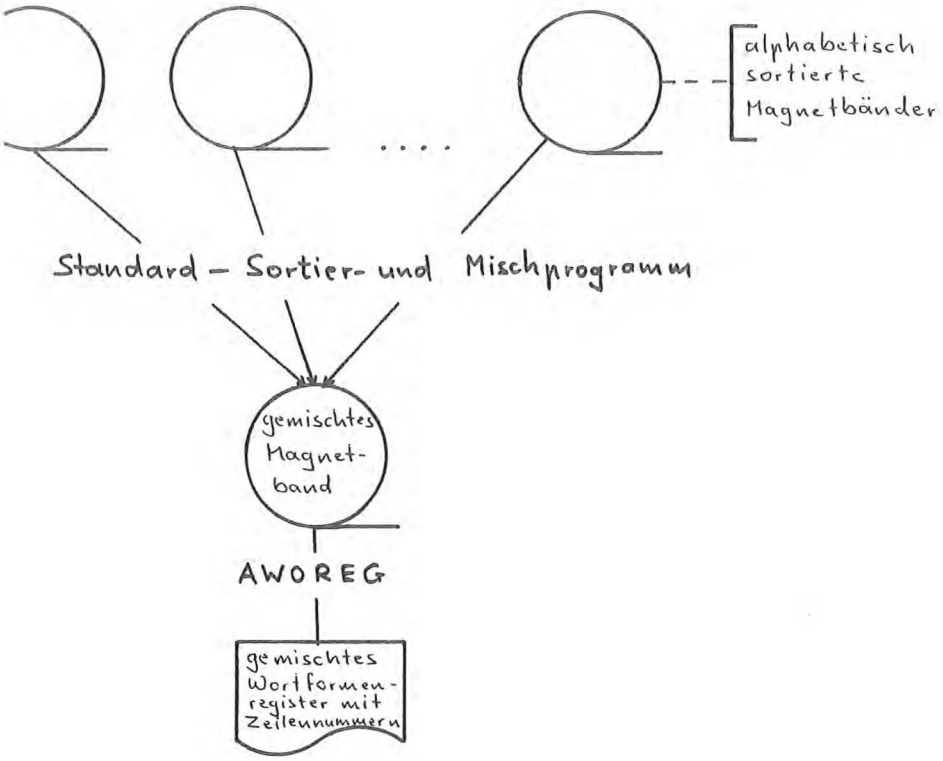
Anhang 5.8.: HÄUFIGKEITSREGISTER

HAEUFIGKEITSREGISTER BERGENGRUEN, DAS TEMPELCHEN SEITE
DATUM:12-1

LAUF. NR.	WORTFORM	HAEUFIGKEIT	ANZ. WF. MIT GL. HF.
1	und	368	1
2	ich	328	1
3	er	168	1
4	die	131	1
5	nicht	130	1
6	das	127	1
7	der	117	
8	in	117	2
9	es	112	1
10	war	108	1
11	auch	95	
12	zu	95	2
13	mir	87	1
14	ein	84	1
15	daß	79	1
16	hatte	77	
17	so	77	2
18	aber	76	1
19	sich	73	1
20	wie	72	1
21	den	64	
22	mich	64	2
23	von	62	1
24	ist	59	1
25	an	56	

Anhang 5.9.: GEMISCHTES WORTFORMENREGISTER MIT ZEILENNUMMERN

vereinfachter Datenflußplan



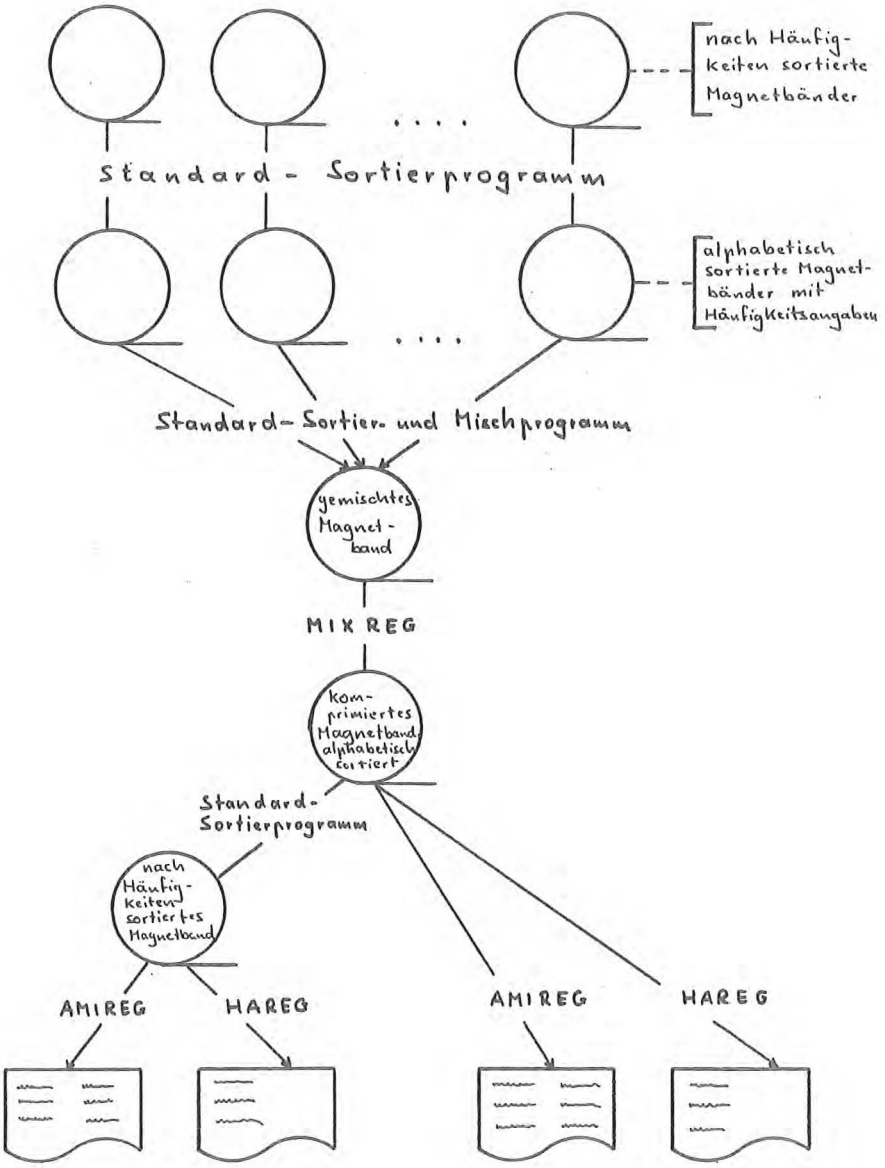
Anhang 5.10.: GEMISCHTES WORTFORMENREGISTER MIT ZEILENNUMMERN (AUS 3 TEXTEN)

ALPHABETISCHES REGISTER MIT HAEUFIGKEITSANGABE GEM. VORL. REGISTER SEITE 14
12:11:73

LAUF. NR.	WORTFORM:	HAEUFIGKEIT: TEXT			ZEILENINDEX:								GESAMT ANZAHL
000227	acht	LBT	*	1	* 005740								
	acht	ZB1	*	16	* 004140	006877	008460	009150	012531	017120	027660	027670	
					* 027690	028010	044570	044710	051280	055460	059900	060120	(00017)
000228	achte	ZB1	*	5	* 013020	028730	039820	040471	040650				(00005)
000229	achten	ZB1	*	1	* 015240								(00001)
000230	achtgeben	LBT	*	1	* 006710								(00001)
000231	adoptieren	ZB1	*	1	* 063470								(00001)
000232	af_	ZB1	*	3	* 013180	043720	044420						(00003)
000233	agam	WHN	*	1	* 016140								(00001)
000234	ago	WHN	*	1	* 016150								(00001)
000235	ahmt	WHN	*	1	* 002860								(00001)
000236	ahnen	WHN	*	1	* 004270								(00001)
000237	ahnte	ZB1	*	1	* 000360								(00001)
000238	ahnten	ZB1	*	1	* 033200								(00001)
000239	ahnungslos	ZB1	*	3	* 005860	015660	020180						(00003)
000240	aktiv	ZB1	*	2	* 028430	040840							(00002)
000241	aktiver	ZB1	*	3	* 028320	028910	040010						(00003)
000242	akute	ZB1	*	1	* 005662								(00001)
000243	akzeptierten	WHN	*	1	* 001930								(00001)
000244	alarmiert	LBT	*	1	* 010830								
	alarmiert	ZB1	*	3	* 007350	024020	035830						(00004)
000245	alarmierte	ZB1	*	5	* 015630	019070	045800	055800	063330				(00005)
000246	alarmierten	ZB1	*	2	* 002730	030930							(00002)
000247	alberne	LBT	*	1	* 011110								(00001)
000248	alias	ZB1	*	1	* 053630								(00001)

Anhang 5.11.: GEMISCHTE REGISTER OHNE ZEILENNUMMERN

vereinfachter Datenflußplan



Anhang 5.12.1.: Gemischtes Häufigkeits-
register

HAEUFIGKEITSREGISTER MIX SEITE 1
DATUM:12-11-7

LAUF. NR.	WORTFORM	HAEUFIGKEIT	ANZ. WF. MIT GL. HF.
1	die	1778	1
2	der	1718	1
3	und	1334	1
4	in	1066	1
5	den	654	1
6	das	649	1
7	ich	643	1
8	nicht	543	1
9	zu	521	1
10	von	519	1
11	mit	517	1
12	er	511	1
13	sich	483	1
14	ein	439	1
15	sie	420	1
16	es	406	1
17	ist	394	1
18	auf	375	1
19	dem	365	1
20	im	359	1
21	daß	346	1
22	eine	344	1
23	für	338	1
24	war	337	1
25	als	321	1
26	auch	312	1

Anhang 5.12.2.: Alphabetisches gemischtes Register ohne Zeilennummern

GEMISCHTES REGISTER DER GESCHRIEBENEN SPRACHE SEITE 7

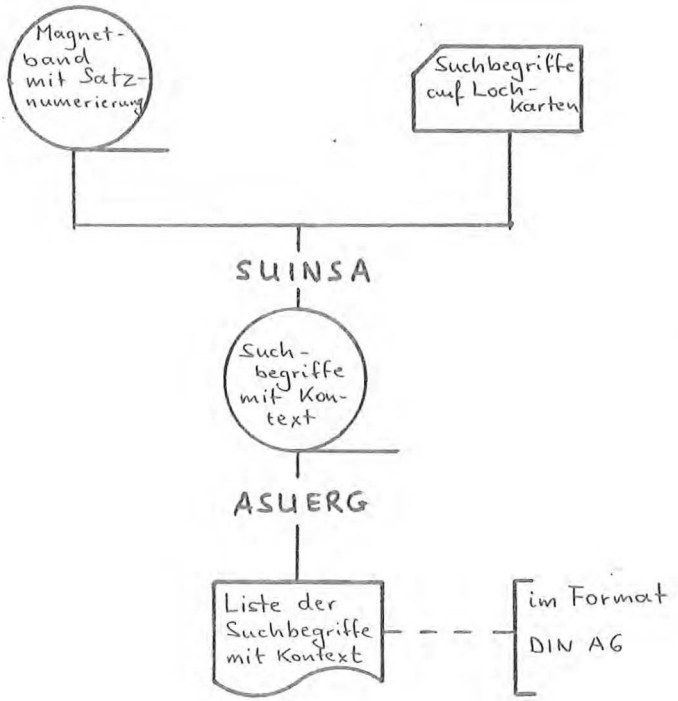
LFNR.	WORTFORM	HAEUFIGK.		LFNR.	WORTFORM	HAEUFIGK.
000601	automatisch	1	**	000651	beendet	1
000602	bändigen	1	**	000652	befördert	1
000603	bäuerlichen	1	**	000653	befürchten	1
000604	böse	2	**	000654	befangen	1
000605	bösen	1	**	000655	befassen	1
000606	böser	1	**	000656	befehlen	1
000607	büßen	1	**	000657	befestigte	1
000608	büffelte	1	**	000658	befiel	1
000609	backt	1	**	000659	befinden	1
000610	bald	9	**	000660	befindlichen	1
000611	baldigen	1	**	000661	befragte	1
000612	banalen	2	**	000662	befreien	1
000613	banden	1	**	000663	befreit	1
000614	barfuß	1	**	000664	befreunden	1
000615	barg	1	**	000665	befreundet	1
000616	bat	5	**	000666	befriedigende	1
000617	baue	1	**	000667	befriedigte	1
000618	bauen	4	**	000668	befruchtet	1
000619	baut	1	**	000669	begünstige	1
000620	bayerische	1	**	000670	begütigt	1
000621	bayerischen	1	**	000671	begabt	1
000622	bayerischer	2	**	000672	begangen	1
000623	bayrische	1	**	000673	begann	1
000624	bayrischem	1	**	000674	begannen	1
000625	bayrischen	1	**	000675	begegne	1
000626	beantrage	1	**	000676	begegnen	1
000627	beantragt	1	**	000677	begegnet	1
000628	beantragte	1	**	000678	begehen	1
000629	beantworten	1	**	000679	begeistert	1
000630	beantwortet	1	**	000680	begeisterter	1
000631	beaten	1	**	000681	beginnen	1
000632	beauftragt	2	**	000682	beginnenden	1
000633	bedacht	2	**	000683	beginnt	1
000634	bedankte	2	**	000684	beglückt	1
000635	bedauerlich	1	**	000685	beglückwünscht	1
000636	bedauerte	1	**	000686	begleiten	1
000637	bedenke	2	**	000687	begleitet	1
000638	bedeute	1	**	000688	begnügen	1
000639	bedeuten	2	**	000689	begnügte	1
000640	bedeutet	9	**	000690	begonnen	1
000641	bedeutete	2	**	000691	begrüßte	1
000642	bedeutsam	1	**	000692	begründen	1
000643	bedient	1	**	000693	begründet	1
000644	bedrängen	1	**	000694	begründete	1
000645	bedrückt	1	**	000695	begraben	1
000646	bedrohlich	1	**	000696	begreiflich	1
000647	bedroht	1	**	000697	begreiflicherweise	1
000648	beeindruckt	1	**	000698	begrenzt	1
000649	beeinflußt	2	**	000699	begrenzte	1
000650	beeinflussen	1	**	000700	behüten	1

RUECKLAEUFIGES REGISTER MIT HAEUFIGKEITSANGABE LB1,DAS TEMPELCHEN SEITE 56
12:11:73

LAUF. NR.	WORTFORM: TEXT		HAEUFIGKEIT:	ZEILENINDEX:				GESAMT ANZAHL
000720	MHE	Reichsflagge	* 1 *	108730				(00001)
000721	MHE	Handelsflagge	* 1 *	108910				(00001)
000722	ZB1	Hausflagge	* 1 *	023570				(00001)
000723	LFH	Flagge	* 1 *	065230				
	WGS	Flagge	* 2 *	023880	024880			
	WPE	Flagge	* 1 *	060710				(00004)
000724	WGS	Ndegge	* 4 *	061680	063400	066480	067170	(00004)
000725	LGB	Hegge	* 1 *	069800				(00001)
000726	LSO	Bulldogge	* 1 *	123000				
	ZB2	Bulldogge	* 2 *	008800	048560			(00003)
000727	LGB	meschugge	* 1 *	066880				(00001)
000728	LSO	mäßige	* 1 *	124950				
	WBM	mäßige	* 2 *	013220	013230			(00003)
000729	LGB	routinemäßige	* 1 *	210510				(00001)
000730	LGB	gleichmäßige	* 2 *	047460	056290			(00002)
000731	WHK	handbuchmäßige	* 2 *	030670	030670			(00002)
000732	MHE	parteimäßige	* 1 *	023290				(00001)
000733	WJA	zweckmäßige	* 1 *	005600				(00001)
000734	LGB	regelmäßige	* 1 *	214960				
	LJA	regelmäßige	* 1 *	052570				
	MHE	regelmäßige	* 1 *	006860				
	WPE	regelmäßige	* 2 *	106670	129400			(00005)
000735	ZB2	unregelmäßige	* 1 *	027880				(00001)
000736	LJA	mittelmäßige	* 1 *	020410				
	WBO	mittelmäßige	* 1 *	079480				(00002)

Anhang 5.14.: SUCHBEGRIFFE MIT KONTEXT : Programm ASULIS

vereinfachter Datenflußplan Unterprogramme SUISA und ASUERG



Anhang 5.15.: SUCHBEGRIFFE MIT KONTEXT

Karteikartenformat

```
I                                                                I
-------------------------------------------------------------------
I SATZNR. 000257  TEXTSCHL. LBT          LFD. NR. 000079         I
I VORGABE  ab                                                    I
I                                                                I
I  er wollte mir seine Ungeduld nicht zeigen , aber ich          I
I spürte sie .                                                   I


I                                                                I
```

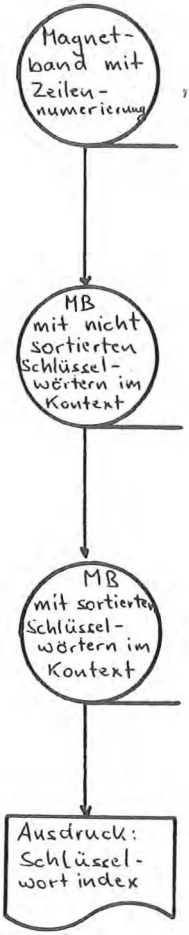
```
I                                                                I
-------------------------------------------------------------------
I SATZNR. 000258  TEXTSCHL. LBT          LFD. NR. 000080         I
I VORGABE  gut                                                   I
I                                                                I
I  er war gut ausgeruht , und auch der Husten schien ihn         I
I nicht mehr zu plagen .                                         I


I                                                                I
```

```
I                                                                I
-------------------------------------------------------------------
I SATZNR. 000263  TEXTSCHL. LBT          LFD. NR. 000081         I
I VORGABE  ig                                                    I
I                                                                I
I  das hätte mich in große Schwierigkeiten gestürzt , und        I
I ich konnte mir nicht vorstellen , wie ich mit ihnen            I
I fertig werden sollte , und so bat ich Gott , er möge es        I
I doch einrichten , daß Jerome kein Pferd von mir                I
I verlangte .                                                    I
```

Anhang 5.16.: SCHLÜSSELWORTINDEX : Programm SOREG

vereinfachter Datenflußplan

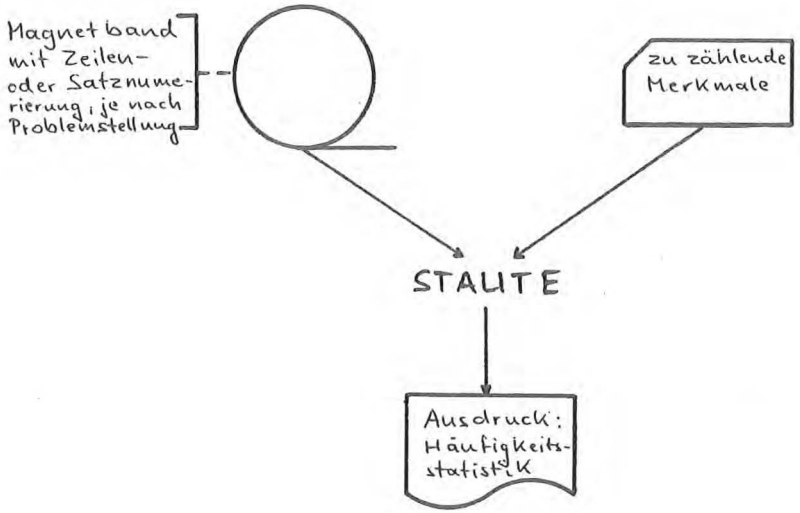


SCHWORINDEX TEXTBEZEICHNUNG: KWIC - LAT SEITE035

TEXT	SCHLUESSELWORT	TEXT
ch genau " . 000690	" ach , Kindchen ,	du sprichst von dieser Veränderung ! . 000700 nun , als sie vorgenommen wurde
her . verzeih mir schon , 010310	Kindchen ,	du weißt , ich werde leicht müde , und 010320 dann bin ich noch vergeßlicher a
die Schneiderinnen und Modistinnen , 010390		du weißt ja , wie das geht . 010400 aber dann bin ich viel , viel später
. das war 010190 natürlich lächerlich , aber		du weißt wohl , wie wir 010200 von der älteren Generation waren : immer , wenn
Hinweis auf Jerome liegen , 009750	das wirst	du zugeben , und ich wußte nur nicht , 009760 wie er zu deuten war . 009770
nd man selten ; wie das jetzt ist , das wirst		du 001340 besser wissen als ich . 001350 Großpapa liebte die Polen nich
Mädchen zu sein pflegen , um die man anhält .		du 001680 warst , glaube ich , zwanzig , nicht wahr ? . mir scheint , 001690
durch die Sümpfe von Stary Dwor zu nehmen .		du 007510 weißt ja , daß Onkel Kostja nichts so lassen konnte , 007520 wie es
bei sich hatten . 007830	das war eigentlich	dumm , denn mit dieser Lichtschein 007840 warnten sie doch jeden , der nicht m
006710 sehr achtgeben , daß nicht irgendeine		dumme Kleinigkeit 006720 in der Kammer zurückblieb , und sei es auch 006730 n
ren gezogen , und jetzt 010940 muß ich diese		dummen Jungen belehren , wie ein 010950 alter Schulmeister ! " . 010960
etrockneten tiefen Brunnen 004870 gefallen ,		dunkel , leer , ohne Laut . 004880 noch einmal wurde er von einem heftig
002410 er hatte schöne , geistvolle Augen ,		dunkelbraun und 002420 von sehr lebhaftem Ausdruck . das alles habe ich 002430
als wenn Sie versuchen 002920 wollten , sich		durch den Park davonzuschleichen . 002930 sicher haben Sie Hunger ? " . 002940
. gern ging sie 000300 am Arm ihrer Enkelin		durch den Park und erzählte 000310 von Vergangenem . wie es bei alten Leuten z
, und 007900 dazwischen blitzten die Sterne		durch die Baumkronen . 007910 jeder Stern , jeder Vogelruf schien mir Bedeutun
o voll , und wie wir da nebeneinander 007950		durch die Dunkelheit wanderten , da kam es mir 007960 vor , als würde und dürf
ntschied sich zuletzt dafür , den Weg 007500		durch die Sümpfe von Stary Dwor zu nehmen . du 007510 weißt ja , daß Onkel Kos
gab es noch nicht ,		durch die 007570 Sümpfe führten ein paar Knüppelwege , aber wer die 007580 ni
eifend , weil Schönheit 002380	und Adel erst	durch eine solche Verfüllung von 002390 Schmutz und Verwahrlostheit hindurchle
ausgestorben war , 000150	kam es schließlich	durch Heirat an den Kollegienassessor 000160 Tschaikin . die Großmutter seiner
t schicken 007270	können , daß Sie glücklich	durchgekommen sind ? " 007280 fragte ich . " mir schreiben ? " . 007290
nd gepflegt . dann 004300	sollte das Kloster	durchsucht werden , aber unmittelbar 004310 vorher haben sie eine Warnung beko
rigkeiten mit sich brachte ! . und es 006680		durfte doch niemand merken , daß ich plötzlich nicht 006690 mehr von Bijou beg
enqung das einzige gewesen . 001290	hier war	eben kein rechter Boden für diese 001300 Geschichten , in unserem Kreise gab e
nem 011300 eigenen Schicksal , und an Jerome		eben nur insofern , 011310 als er zu diesem Schicksal gehörte . 011320 01133
un , ein andermal , ich merke schon , 010290		eben will mir der Name nicht einfallen , und dann 010300 gehört es wohl auch n
meinnicht wuchsen ; 009450 nun , da habe ich		eben Vergißmeinnicht gepflückt , 009460 ohne mir etwas dabei zu denken . da ha
delte oder um jemand anders , das 010470 war		ebenso ungewiß wie die Geschichte mit der Medaille , 010480 und manchmal möcht
sei alles übrige schon abgemacht . 004940 "		ebensowenig wie Licht machen . sonst aber 004950 kann Ihnen dort nichts zustoß
Dinge ereignen . ich bekreuzte mich , 008870		ehe ich hinüberging , und die Knie haben mir gewankt . 008880 Papa saß an sein
t , 010250 eine entfernte Cousine oder einen		ehemaligen Tänzer . 010260 und oft geschah das auch . habe ich dir schon 01027
einandergesetzt 008360	hatte , wie ein Pferd	eher Gefahr 008370 bringen werde als Nutzen . 008380 Jerome hörte mir z
gen ? . ich fand ja auch , daß er ein 009110		ehrenhafter und guter Mensch war , und mein Herz 009120 war so zerrissen , daß
diesen 000500 Gehalt zu erkennen und für das		eigene Dasein 000510 fruchtbar zu machen . 000520 eines Vormittags , al
ch 010980 merkte daran , daß er nur an seine		eigenen Angelegenheiten 010990 gedacht hatte . ach so , den Polen ? . nein , O
n nicht an Jerome , sondern an meinem 011300		eigenen Schicksal , und an Jerome eben nur insofern , 011310 als er zu diesem
te ist es gar nicht , und interessant 010760		eigentlich auch nicht , nur daß mir dabei allerhand 010770 Gedanken gekommen s
n , die sie bei sich hatten . 007830	das war	eigentlich dumm , denn mit diesem Lichtschein 007840 warnten sie doch jeden ,
bezaubernde Art , zu sprechen , aber 006470		eigentlich hat er mir doch von sich selber fast 006480 nichts erzählt , das er
n - nein , 011710 angetragen habe ich es ihm		eigentlich nicht , aber 011720 entgegengetragen , verstehst du , stillschweige
zu haben in einem 005480 Alter , in dem man		eigentlich noch keine Geheimnisse 005490 bewahren kann ! . das ist , als würde
tgehen können ? . 005800 ja , hätte es nicht		eigentlich so fortgehen müssen ? . 005810 ich versuchte , ihm die Meinung aufz
anzen Sache überhaupt wenig gemerkt , 001280		eigentlich war die Brückensprengung das einzige gewesen . 001290 hier war eben
u helfen 006980 und so sehr ich hierin meine		eigentliche Lebensaufgabe 006990 zu sehen meinte , - ich fürchtete mich doch O
dann 006440 fiel mir doch nichts Gescheites		ein . und er ? . ja , er 006450 hatte wohl eine bezaubernde Art , zu sprechen
Jugend bilden sie es sich 012040 allenfalls		ein , aber nachher ist da immer etwas anderes , 012050 und dies Andere ist das
Wissenschaften " , und dann fiel Mama 009040		ein , schluchzte ein wenig und meinte , er könnte es 009050 weit bringen , und
en . ich schloß mich in meinem Zimmer 007780		ein , und ein wenig später bin ich aus dem Fenster 007790 gestiegen . 007800
O wir im Auslande reisten , bildeten wir uns		ein , wir 010220 wüßten jemanden von zu Hause treffen , einen Gutsnachbarn 010

Anhang 5.18.: HÄUFIGKEITSSTATISTIK : Programm STAUTE

vereinfachter Datenflußplan



Anhang 5.19.: HÄUFIGKEITSSTATISTIK

```
BEZEICHNUNG DES TEXTES:
LAT,DAS TEMPELCHEN
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
UNTERSUCHUNG DES GESAMTTEXTES NACH HAEUFIGKEITEN
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

ES WURDE KEINE AUSWAHL MIT DEM ZUFALLSZAHLENGENERATOR GETROFFEN

ANZAHL DER VERARBEITETEN ZEILEN:  435

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER   SAETZE IM TEXT:     434
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER  WOERTER IM TEXT:    8468
DAVON SUBSTANTIVE:    1200=14,17%
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER  ZEICHEN IM TEXT:   43209
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER SONDERZ. IM TEXT:    1569
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER   VOKALE IM TEXT:   15793
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER  KONSON. IM TEXT:   25847
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANZAHL DER   ZAHLEN IM TEXT:       0
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

MINIMALE-,MAXIMALE- UND DURCHSCHNITTL. ANZ. VON WOERTERN UND ZEICHEN IN SAETZEN

    GESAMTTEXT
    ----------
FUER WOERTER:          MINIMUM:   1     MAXIMUM:  74     DURCHSCHNITT:  19
FUER ZEICHEN:          MINIMUM:   7     MAXIMUM: 387     DURCHSCHNITT:  99
```

HAEUFIGKEIT DER EINZELNEN BUCHSTABEN
UND DEREN PROZENTUALER ANTEIL AN DER
GESAMTHAEUFIGKEIT ALLER BUCHSTABEN

ZEICHEN	HAEUF.	PROZ.
A	2664	6.40%
B	759	1.82%
C	1750	4.20%
D	1947	4.68%
E	6654	15.98%
F	500	1.20%
G	1178	2.83%
H	2610	6.27%
I	3393	8.15%
J	186	0.45%
K	543	1.30%
L	1405	3.37%
M	1286	3.09%
N	4426	10.63%
O	960	2.31%
P	246	0.59%
Q	5	0.01%
R	2661	6.39%
S	2267	5.44%
T	2253	5.41%
U	1521	3.65%
V	287	0.69%
W	837	2.01%
X	2	0.00%
Y	5	0.01%
Z	432	1.04%
Ä	209	0.50%
Ö	107	0.26%
Ü	280	0.67%
ß	267	0.64%

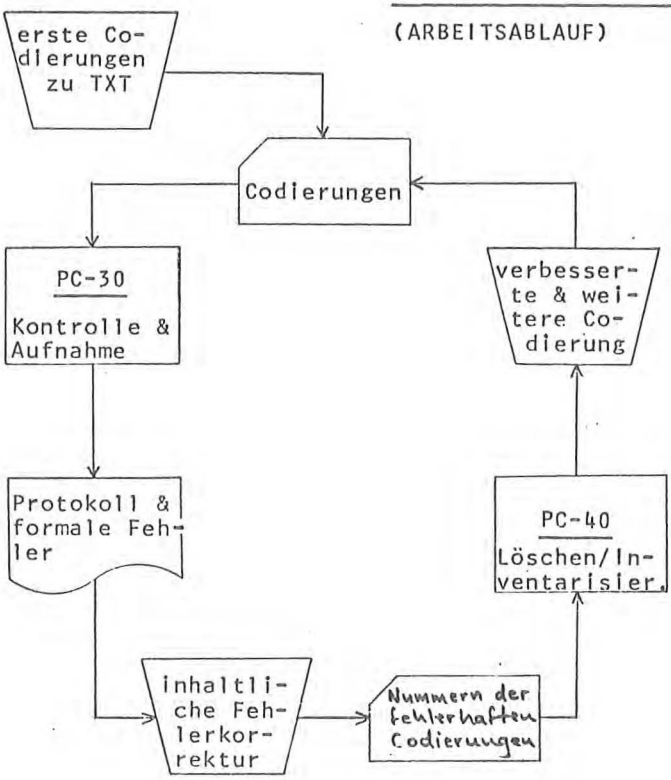
HAEUFIGKEIT DER EINZELNEN SONDERZEICHEN
UND DEREN PROZENTUALER ANTEIL AN DER
GESAMTHAEUFIGKEIT ALLER SONDERZEICHEN

ZEICHEN	HAEUF.	PROZ.
.	437	27.85%
<	0	0.00%
(	3	0.19%
+	0	0.00%
|	0	0.00%
&	0	0.00%
!	26	1.66%
$	0	0.00%
*	0	0.00%
)	3	0.19%
;	13	0.83%
	0	0.00%
-	22	1.40%
/	0	0.00%
§	0	0.00%
[	0	0.00%
]	0	0.00%
,	922	58.76%
%	0	0.00%
_	0	0.00%
>	0	0.00%
?	35	2.23%
:	8	0.51%
#	0	0.00%
@	0	0.00%
'	92	5.86%
=	0	0.00%
"	8	0.51%

Anhang 5.20.: PARALLELCODIERUNG

ERFASSEN, LOESCHEN UND INVENTARISIEREN VON CODIERUNGEN.

(ARBEITSABLAUF)



Anhang 5.21.1.: M e r k m a l s v o r r a t

Die nachfolgenden Beispielsausdrucke gehen von Texten der *gesprochenen* Sprache aus, da Parallelcodierungen bisher nur hierfür durchgeführt wurden. Die Vorgehensweise bei Texten der geschriebenen Sprache ist jedoch grundsätzlich dieselbe.

Abbildungen dazu siehe Seiten 94 bis 98

Anhang 5.21.2.: C o d i e r t e M e r k m a l e m i t F e h l e r k e n n z e i c h n u n g

Abbildungen dazu siehe Seiten 119 bis 121

Anhang 5.21.3.: S t r u k t u r b a u m

Abbildung dazu siehe Seite 185

Anhang 5.21.4.: P r o t o k o l l e f ü r e i n e n S u c h b e g r i f f

Abbildungen dazu siehe Seiten 180 bis 182

Berthold Epp

P A R A L L E L C O D I E R U N G

Das Verfahren und seine Anwendung

Ich muß entweder ein System
erschaffen oder Sklave des-
jenigen eines anderen werden.
Ich will nicht räsonnieren
oder vergleichen; meine Auf-
gabe ist, Neues zu schaffen.

William Blake

INHALTSVERZEICHNIS Seite

Seite

Seite

VORWORT

Die vorliegende Schrift ist das Ergebnis einer Arbeit, die im Auftrag des Instituts für deutsche Sprache durchgeführt wurde. Dem Institut steht eine umfangreiche Sammlung deutschen Gegenwartsschrifttums zur Verfügung[1].

"Mannheimer Korpus":

eine Sammlung von Texten der geschriebenen Sprache (Romane, Erzählungen, Memoiren, populärwissenschaftliche Schriften, Trivialliteratur und Zeitungstexte) mit einem Umfang von zirka 2.2 Millionen Wörtern.

"Freiburger Korpus":

eine Sammlung von Texten der gesprochenen Sprache, die in ihrem endgültigen Umfang zirka 600·000 Wörter umfaßt.

"Bonner Korpus":

eine Sammlung von zeitlich parallelen Zeitungstexten aus der DDR der Bundesrepublik mit zirka 1.2 Millionen Wörtern.

Diese Texte liefern das Material für wissenschaftliche Forschungsarbeiten am Institut.

Die Bearbeitung der Texte mit Hilfe der elektronischen Datenverarbeitung findet dort ihre Grenzen, wo die in ihnen enthaltenen Informationen syntaktischer oder semantischer Art nicht maschinell herauslösbar sind, um dann Grundlage für spezielle Untersuchungen sein zu können. Da die Entwicklung von Analyseprogrammen[2] das Ergebnis einer Untersuchung voraussetzt, können solche automatische Verfahren nicht Hilfsmittel für lau-

fende Forschungsarbeiten sein. Auch legen sie eine bestimmte Modellvorstellung zugrunde und sind nicht anwendbar, wenn immer ein anderer theoretischer Überbau an deren Stelle tritt. So wurde bereits im Jahre 1968 von Alex Ströbl[3] ein Verfahren formuliert, das in die Lage setzen soll, relevante Informationen einem Text beizuordnen, um dann den Einsatz der elektronischen Datenverarbeitung zu ermöglichen. Die ersten Ansätze zur Verwirklichung des Verfahrens hatten nur begrenzten Erfolg, insbesondere konnte keine Möglichkeit einer universellen Zuordnung von Merkmalen zum Text gefunden werden.

Die nun vorliegende Konzeption und ihre Verwirklichung durch ein Programmsystem schließlich hat aus diesen Ansätzen nur die zugrundeliegende Idee übernehmen können: der für die maschinelle Verarbeitung aufbereitete Text soll unverändert bleiben. Dafür aber gab es eine Reihe von Gesichtspunkten, die teilweise von vorneherein formuliert, teils aber auch erst während der Bearbeitung deutlich wurden.

- Es galt, eine Möglichkeit zu finden, die das Verfahren selbst unabhängig sein läßt von einer linguistischen Modellvorstellung.
- Innerhalb einer Modellvorstellung sollen mehrere Betrachtungsebenen zulässig sein.
- Die Informationsverschlüsselung soll transparent und effizient zugleich sein.
- Auf Textsatzebene muß jeder wie auch immer geartete Bezug zum Text möglich sein.
- Das Verfahren soll sich nicht auf die Verarbeitung von laufendem Text beschränken, sondern auch für andere Probleme (z. B.: Lexikonverarbeitung) eingesetzt werden können.

Das Bestreben, einfache und unproblematische Anwendung, sowie deutliche Lesbarkeit zu erzielen, stand dabei im Vordergrund.

Für die Programmierung stand eine Rechenanlage SIEMENS 4004/35 zur Verfügung, auf der sämtliche Programme entwickelt und getestet wurden. Aus Gründen der anzustrebenden Kompatibilität, nicht nur in den Daten, sondern auch in den Programmen selbst, wurde als Programmiersprache FORTRAN gewählt, und zwar in einem Sprachumfang, wie er auf den meisten Anlagen entsprechender Größenordnung verfügbar ist. Wenn auch das eine oder andere Programm bei der Implementierung auf einer Rechenanlage mit anderer Speicherstruktur modifiziert werden muß, die zentrale Bitketten-Verarbeitungsprogramme sind kompatibel und wurden auch unter diesem Gesichtspunkt getestet. Die Aufgabe erhielt dadurch einen besonderen Reiz, daß eine Programmiersprache, die von ihrer Konzeption ganz auf mathematisch-technische Fragestellungen orientiert ist, in der nichtnumerischen Datenverarbeitung eingesetzt wurde. Die Verwendung einer höheren Programmiersprache schließlich bietet zudem für den Anwender die Möglichkeit, die vorliegenden Programme zur Parallelcodierung ohne Schwierigkeit um eigene Routinen zu ergänzen.

Mein Dank gilt an dieser Stelle allen Mitarbeitern der Forschungsstelle Mannheim der Abteilung Linguistische Datenverarbeitung im Institut für deutsche Sprache, die alle jederzeit meinen Nöten und Fragen offen gegenüberstanden. Er gilt aber auch den Mitarbeitern in der Forschungsstelle Freiburg unter Leitung von Herrn Prof. Dr. H. Steger, insbesondere Frau Schoenthal und Herrn Wilbs. Viele nützliche Anregungen habe ich von dort erhalten, so waren sie auch bereit, mir schon für die Entwicklung der Programme umfangreiches Testmaterial zur Verfü-

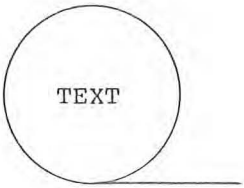
gung zu stellen. Mit ihrer freundlichen Erlaubnis kann ich im Rückgriff auf dieses Material alle Einzelschritte mit Beispielen belegen.

Wilhelmsfeld, im August 1972

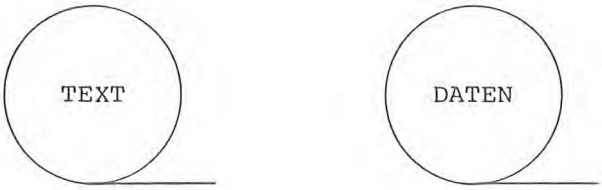
Berthold Epp

O. EINLEITUNG

Wir betrachten Texte, die einer maschinellen Verarbeitung zugänglich sind und, wie im Falle des Mannheimer Korpus, als Folge von Sätzen[4] auf Magnetbändern vorliegen.



Diese Texte werden unter linguistischen Aspekten analysiert. Die so konkretisierten Informationen, seien sie maschinell oder manuell erzeugt, sollen für eine weitere maschinelle oder manuelle Analyse verfügbar sein. Dazu werden sie nicht in den bereits bestehenden Text eingefügt, sondern als eigenständiger Datenpool auf Magnetbändern niedergelegt.

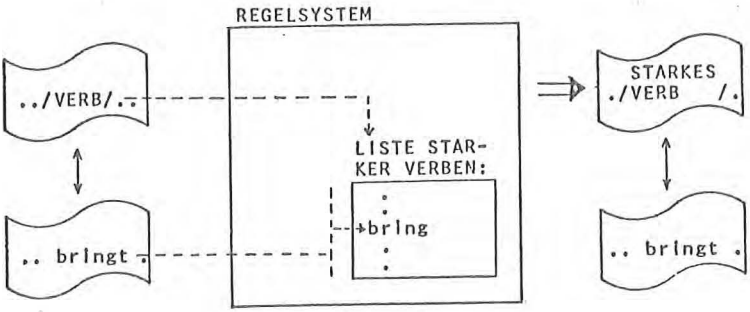


Aus der Parallelität von Text und zugehörigen Daten entsprang der Name für das Projekt: Parallelcodierung.

Die syntaktische oder semantische Analyse von

Texten setzt in jedem Falle ein der Analyse zugrundeliegendes Modell voraus. Da es kein allgemein gültiges Modell hierzu gibt, muß jede konkrete Analyse auf dem Modell basieren, dessen sich ein Bearbeiter bedient. So kann die Arbeit des einen Ergebnisse liefern, die von den Ergebnissen eines anderen trotz gleichen Arbeitsobjektes abweichen, oder aber, falls ähnliche Ergebnisse erzielt sind, können sich die Wege, auf denen sie erreicht wurden, stark unterscheiden.

Dieser Sachverhalt stellt an das zu realisierende System zwei Forderungen. Zum einen muß das System Parallelcodierung so flexibel sein, daß es unabhängig ist von dem einer Textanalyse jeweils zugrundeliegenden Modell. Zum anderen soll die einmal erfaßte Information umdeutbar sein, damit sie auch unter einer anderen Modellvorstellung möglichst weitgehend verwertbar ist. Damit wird verlangt, daß die gespeicherte Information so transparent aufbereitet ist, daß auch einzelne Teile aus ihr entnommen werden können, um so beispielsweise mit einem ergänzenden Regelsystem und dem zugehörigen Text Ausgangsmaterial für eine weiterführende maschinelle Analyse zu sein.



So wurde die Parallelcodierung als ein Teilnehmersystem entwickelt, bei dem der einzelne Teilnehmer sein eigenes Begriffsspektrum (= die Menge der von ihm für eine Analyse als relevant erachteten Merkmale) verwendet und sich nur dem formalen Mechanismus des Systems unterordnet.

Für die eigentliche Informationsverschlüsselung wurde ein Codierungsverfahren verwendet, das den Zugriff zu jeder beliebigen Teilinformation erlaubt und das zugleich dem Arbeitsmodus einer digitalen Rechenanlage angepaßt ist.

1. FORMALISIERUNG DER TEXTANALYSE

1.1 Textanalyse als Abbildung

Bei der linguistischen Analyse von Texten gehört die Segmentierung der Texte in Teilsequenzen und die Feststellung der Klassenzugehörigkeit der ermittelten Teilsequenzen zu den notwendigen Grundoperationen. Diese Operationen der Teilung und Klassifikation lassen sich auf den verschiedensten Analyseebenen und unter den verschiedensten theoretischen Gesichtspunkten durchführen.

Beispiel: Legen wir für eine linguistische Theorie zur Textanalyse eine einfache Phrasenstrukturgrammatik zugrunde, dann wären durch die Ebenen einer phrasenstrukturellen Satzbeschreibung zugleich denkbare Analyseebenen unter dieser Modellvorstellung gegeben:

- Ebene lexikalischer Formative
- Verschiedene Ebenen höherer Satzkonstituenten
- Satzebene

In der Regel sind vereinbarten Analyseebenen Teilsequenzen eines Textes zugeordnet. In unserem Zusammenhang interessiert uns an dieser Stelle nur der rein formale Aufbau dieser Sequenzen, wie er sich optisch dem Leser darstellt, oder als Aufeinanderfolge von Zeichen von der Maschine erkannt wird.

Für das obige Beispiel lassen sich solche Sequenzen sofort angeben:

- eine zusammenhängende Folge von Textzeichen, die kein Leerzeichen enthält (Lexeme, Morpheme)
- eine zusammenhängende Folge von Textzeichen, die kein Leerzeichen enthält und durch zwei Leerzeichen eingeschlossen ist (Textwort)
- eine Folge zusammenhängender Textwörter
- eine geschachtelte Folge von Textwörtern oder Folgen von Textwörtern, bei denen der formale Zusammenhang der Textwörter verloren gegangen ist.

In jedem Falle aber läßt sich eine Begrenzung der Sequenzen angeben. Sie zu finden, also die Segmentierung des Textes vorzunehmen, ist Bestandteil der Textanalyse.

Wir nehmen an, es stehe uns eine vollständige und erschöpfende linguistische Theorie zur Textanalyse zur Verfügung. Dann definiert diese (fiktive) Theorie Ebenen ähnlich den oben angegebenen. Zugleich ist es dem Linguisten möglich, einen vorgegebenen Text auf jeder so erklärten Ebene in Teilsequenzen zu zerlegen. Diese Theorie definiert nun weiter auf jeder Ebene Informationsklassen, etwa die Klassen der Information über Person, Numerus, Genus, Kasus etc., wobei die einzelnen Klassen beschrieben werden können durch die Aufzählung aller ihrer Elemente. So würde beispielsweise die Informationsklasse "Kasus" beschrieben durch die Elemente Nominativ, Genitiv, Dativ, Akkusativ. Ein Element einer Informationsklasse ist schließlich irgendeine nicht weiter zerlegbare begriffliche Einheit, deren Inhalt durch das der Textanalyse zugrundeliegende Modell bestimmt ist. Derartige Elemente nennen wir im

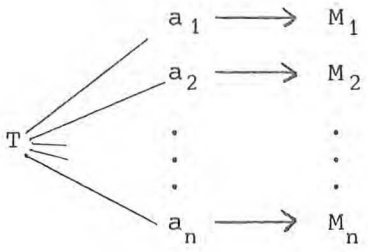
Rahmen der Parallelcodierung Merkmale.

Gehen wir davon aus, daß unsere fiktive Theorie eine gewisse Anzahl n von Ebenen a_1, a_2, ... a_n erklärt, auf denen eine Textanalyse durchgeführt werden kann, dann nennen wir die Menge aller Merkmale zur Ebene a_i ($1 \leq i \leq n$) den Merkmalsvorrat M_i. Die Analyse eines Textes T auf einer Ebene a_i, die ja nichts anderes ist als die Zuordnung von Teilsequenzen des Textes zu den definierten Merkmalen des Merkmalsvorrats M_i, können wir dann verstehen als eine Abbildung

$$T \xrightarrow{a_i} M_i$$

Der Teil unserer Theorie, der sich auf die Ebene a_i bezieht, bestimmt eindeutig die Segmentierung des Textes auf dieser Ebene und erklärt, welchem Element aus M_i eine Teilsequenz zuzuordnen ist. Insofern dürfte es nicht zu Mißverständnissen führen, wenn wir die Abbildungsvorschrift mit dem Namen der zugehörigen Ebene identifizieren.

Eine umfassende Analyse auf n verschiedenen Ebenen läßt sich dann so verstehen:



Wir setzten bis jetzt die Existenz einer vollständigen und erschöpfenden Theorie zur Textanalyse voraus. Das ist eine Annahme, von der wir nicht ohne weiteres ausgehen können. Insofern müssen wir bezweifeln, ob eine bestehende Theorie, mit der wir aber arbeiten wollen, solche Ebenen a_i und zugehörige Merkmalsvorräte M_i definiert.

Hier müssen zwei Fälle unterschieden werden.

1. Fall: Für wenigstens ein i ist a_i nicht vollständig erklärt. Mit anderen Worten: An einer Textstelle ist es nicht möglich, unter der formulierten Theorie eine Segmentierung des Textes durchzuführen. (Aus dem Definitionsbereich T der Abbildung a_i läßt sich kein Element bestimmen, das mit M_i in Relation gebracht werden könnte.) In diesem Falle scheint es uns notwendig, die Theorie selbst zu überprüfen, denn offensichtlich sind hier Merkmale definiert worden, die entweder widersprüchlich sind oder sich nicht ausreichend gegeneinander abgrenzen, so daß die Bestimmung der Teilsequenz problematisch ist.

2. Fall: M_i ist unvollständig. Es läßt sich zwar eine Segmentierung durchführen, doch es gibt in M_i kein Merkmal, das als zutreffend für die ausgewählte Textsequenz angegeben werden könnte. In diesem Falle bieten sich mehrere Auswege an. Zum einen entscheidet man sich dafür, die Textsequenz unberücksichtigt zu lassen. Im Sinne der Parallelcodierung heißt das, sie wird nicht codiert. Andererseits lassen sich möglicherweise zutreffende Merkmale aus noch zu beschreibenden Subklassen angeben, so daß wenigstens diese Information festgehalten wird. In der Regel aber wird man dieses Phänomen genau analysieren und gegebenenfalls den Merkmalsvorrat der zugehörigen Ebene erweitern.

Zusammenfassend erwartet das System Parallelcodierung von einer linguistischen Theorie zur Textanalyse nur die vollständige Segmentierbarkeit eines Textes auf den einzelnen Ebenen.

Segmentierung und Ermittlung der Klassenzugehörigkeit verstehen wir dann als Abbildung eines Textes in einen Merkmalsvorrat, wobei die Abbildungsvorschrift durch die zugrundeliegende linguistische Theorie bestimmt wird.

Sollen die Texte mit Hilfe der Datenverarbeitung untersucht werden, so kann eine solche Merkmalszuweisung grundsätzlich auf zwei Wegen erfolgen: Entweder ist ein Analyseprogramm zu erstellen, das es erlaubt, automatisch die zur Segmentierung und Klassifikation relevante Information aus dem Text zu erschließen. Der andere Weg besteht darin, daß der Linguist aufgrund seines "Textverständnisses" den Teilsequenzen eines Textes "von Hand" Merkmale zuordnet. Dieses Verfahren ist weniger ökonomisch und einem Analyseprogramm wohl nur dann vorzuziehen, wenn das begrenzte Vorwissen über ein zu untersuchendes Phänomen die Entwicklung eines entsprechenden Algorithmus noch nicht erlaubt. Die zweite Verfahrensweise wird beispielsweise der automatischen Erstellung von Exzerpten, bzw. vorläufigen statistischen Erhebungen im Rahmen laufender Forschungsarbeiten dienen. Insbesondere denken wir aber auch an die Möglichkeit, die zu einer oder mehreren Ebenen einer Analyse manuell erstellte Information in den Informationspool für ein Analyseprogramm zu einer anderen Ebene einzubringen.

1.2 Klassifikation des Urbildbereiches

Für die folgende Betrachtung legen wir die Be-

griffe der naiven Mengenlehre zugrunde, wonach eine M e n g e der Sammelbegriff für gewisse E l e m e n t e ist, die voneinander wohlunterschieden werden können. Wenn a ein Element der Menge A ist, dann schreiben wir: a ϵ A. Eine Menge A heißt T e i l m e n g e einer Menge B, wenn jedes Element aus A auch Element aus B ist. Die L e e r e M e n g e , das ist die Menge, die kein Element enthält, sei mit $\emptyset$ bezeichnet. Die V e r e i n i g u n g zweier Mengen A und B (in Zeichen: A $\cup$ B) enthält alle diejenigen Elemente, die in A oder auch in B enthalten sind. Der D u r c h s c h n i t t zweier Mengen A und B (in Zeichen: A $\cap$ B) enthält die Elemente, die sowohl in A als auch in B enthalten sind. Zwei Mengen heißen d i s j u n k t oder elementfremd, wenn ihr Durchschnitt die leere Menge ist.

Der zur Ebene a_i gehörige Merkmalsvorrat M_i enthält als Elemente alle diejenigen Begriffe, die für ein bestimmtes Analysemodell definiert sind und sich auf bestimmbare Teilsequenzen eines Textes beziehen können.

Sei nun M ein für eine Ebene erklärter Merkmalsvorrat. Dann läßt sich M aufspalten in zwei disjunkte Teilmengen $\overline{M}$ und $\underline{M}$.

$$\overline{M} \cup \underline{M} = M$$

$$\overline{M} \cap \underline{M} = \emptyset$$

Die Elemente von $\overline{M}$ nennen wir die obligatorischen Merkmale und $\underline{M}$ sei die Menge der fakultativen Merkmale. Durch diese Aufspaltung wollen wir eine erste Klassifizierung von M erhalten mit dem Ziel der Rückführung komplexer Merkmalsaus-

prägungen auf ihre atomaren Bestandteile.

Beispiel: Seien "Verbalgruppe in der ersten Person Singular" und "Verbalgruppe im Infinitiv" zwei Merkmalsausprägungen in M, so läßt sich eine Aufspaltung denken von $M = \overline{M} \cup \underline{M}$ und "Verbalgruppe" $\in \overline{M}$.

Zweckmäßigerweise werden wir als obligatorisch diejenigen Merkmale bzw. Bausteine von Merkmalsausprägungen benennen, denen im Rahmen des übergeordneten Modells die gewichtigere Rolle zur Unterscheidung von Phänomenen der sprachlichen Äußerung zufällt.

Seien $m_1, m_2, \ldots m_k$ die als obligatorisch ausgezeichneten Merkmale in M, dann muß gelten: für jede Teilsequenz t eines Textes T unter der Analyse a mit dem zugehörigen Merkmalsvorrat $M = \overline{M} \cup \underline{M}$ gibt es in $\overline{M}$ h ö c h s t e n s e i n m_j $(1 \leq j \leq k)$ derart, daß m_j für t unter a zutrifft und nicht zwei zugleich. Damit wird also gefordert, daß die obligatorischen Merkmale eindeutig definiert werden und sich gegenseitig ausschließen.

Da wir nur die vollständige Segmentierbarkeit eines Textes voraussetzen, kann der Fall eintreten, daß kein m_j für ein gewisses t unter a zutrifft, daß also der theoretische Überbau zu M nicht erschöpfend ist, da sich keine, wenigstens globale Zuordnung treffen läßt. Dem zu begegnen führen wir ein obligatorisches Merkmal e ein, dessen spezielle Aussage in $\overline{M}$ sei: "kein $m_j \in \overline{M}$ trifft zu". Dann ist $\overline{M}$ vollständig beschrieben durch seine Elemente

$$\overline{M} = \{e, m_1, m_2, \ldots m_k\}$$

und wir stellen fest, daß es zu jedem $t \in T$

g e n a u e i n $m \in \overline{M}$ gibt, so daß m für t unter a zutrifft.

Die nach Abtrennen von $\overline{M}$ aus M verbleibenden Informationsträger, zusammengefaßt in der Menge der fakultativen Merkmale $\underline{M}$, sind geeignet, ein Element $m \in \overline{M}$ zu ergänzen und somit die spezielle Ausprägung einer Textsequenz zu beschreiben. Falls $\underline{M}$ nicht leer ist, lassen sich in der Regel in $\underline{M}$ Subklassen bilden derart, daß die Elemente solcher Klassen sich, ebenso wie die Elemente von $\overline{M}$, gegenseitig ausschließen, daß also, falls überhaupt ein Element einer Klasse für die Textsequenz zutrifft, höchstens eines aus der Klasse zutrifft.

Subklassen in diesem Sinn wären beispielsweise die bereits angesprochenen Klassen Numerus, Genus und Kasus. Eine Phrase steht entweder im Singular oder im Plural, beide Merkmale können nicht zugleich zutreffen. Somit wären die Merkmale der Subklasse Numerus die Eigenschaften Singular und Plural.

Ein Element einer Subklasse kann einem oder verschiedenen obligatorischen Merkmalen beigeordnet werden. Falls diese Beiordnung möglich ist und getroffen wird, stellt sie eine Verfeinerung der Analyse dar, eine Notwendigkeit im Rahmen der Parallelcodierung besteht für eine solche erweiternde Zuordnung nicht.

Eine Subklasse in der Menge der fakultativen Merkmale kann möglicherweise nur ein Element enthalten. $\underline{M}$ selbst kann seinerseits leer sein, so daß dann der Merkmalsvorrat nur aus obligatorischen Merkmalen besteht.

Wir stellen zusammenfassend fest: Durch a wird

jeder Teilsequenz t eines Textes T genau ein obligatorisches Merkmal $m \in \overline{M}$ zugeordnet. Falls $M \neq \emptyset$ gibt es möglicherweise ein oder mehrere fakultative Merkmale $b_1, b_2, \ldots$, die durch a den obligatorischen Merkmalen beigeordnet werden. Dabei gilt für die einzelnen b_i, daß sie aus disjunkten Subklassen stammen. Wir erhalten also für

$$T \xrightarrow{a} M$$

die elementweise Abbildungsvorschrift:

$$t \xmapsto{a} (m; b_1, b_2, \ldots)$$

mit $t \in T$ (t eine Textsequenz unter a)

$m \in \overline{M}$ (m obligatorisches Merkmal unter a)

$b_1, b_2, \ldots \in \underline{M}$ (fakultative Merkmale unter a zu m)

1.3 Voraussetzungen für den Einsatz der Parallelcodierung

Die bisherigen Betrachtungen sollten in aller Ausführlichkeit die notwendigen Voraussetzungen für den Einsatz der Parallelcodierung aufzeigen:

- Das einer Analyse zugrundeliegende Modell erklärt Ebenen, auf denen die Textanalyse durchgeführt werden kann. (Für die Anwendung der Parallelcodierung genügt bereits eine Ebene, innerhalb eines Problemkrei-

ses sind maximal 10 Ebenen zulässig.)

- Zu jeder Ebene gibt es eine Menge von Merkmalen (Merkmalsvorrat), in der jedes Merkmal von jedem anderen genau abgegrenzt ist. (Das ist erreichbar durch die Reduktion auf atomare Bestandteile.)

- Die Elemente eines Merkmalsvorrats sind klassifizierbar derart, daß nicht zwei Merkmale aus einer Klasse zugleich für die selbe Teilsequenz zutreffen können. (Die terminologische Unterscheidung nach obligatorischen und fakultativen Merkmalen ist Orientierungshilfe und hat in erster Linie programmtechnische Gründe.)

- Ein Text ist wenigstens soweit segmentierbar, wie die Sequenzen mit Merkmalen belegt werden.

2. VORBEREITEN EINER PARALLELCODIERUNG

2.1 Erstellen eines Code-umsetzers

Wir können nun davon ausgehen, daß eine Abgrenzung der Ebenen gegeben ist und zu jeder Ebene ein Merkmalsvorrat existiert. In der Regel sind die Merkmale natürlich-sprachlich formuliert, ihre Abgrenzung ist anhand von Beispielen belegt.

Im Hinblick auf den Einsatz einer Datenverarbeitungsanlage werden nun sämtlichenMerkmale mnemonische Kurzwörter zugeordnet, also Abkürzungen, die möglichst lesbar formuliert sind. Diese Abkürzungen dienen für den gesamten Einsatz der Parallelcodierung als Kommunikationsmittel: durch sie werden den Verarbeitungsprogrammen die zu verschlüsselnden Informationen mitgeteilt, Ergebnisse werden mit ihrer Hilfe ausgedruckt.

Formal ist für die Festlegung von mnemonischen Kurzwörtern zu beachten:

- Sie bestehen aus eins bis vier Zeichen[5], wobei alle Zeichen, mit Ausnahme des Leerzeichens (Blank) " " und des Pluszeichens "+" zugelassen sind.
- Innerhalb einer definierten Ebene müssen alle festgelegten mnemonischen Kurzwörter voneinander verschieden sein.
- Das mnemonische Kurzwort "...." darf nicht explizit erklärt werden. Es wird automatisch jeder Ebene beigeordnet und

übernimmt die Rolle des neutralen Elementes e unter den obligatorischen Merkmalen zu den Ebenen und hat die Bedeutung: "kein obligatorisches Merkmal trifft zu".

Ist der Einsatz der Parallelcodierung geplant, wird die Bereitstellung eines Codeumsetzers angefordert. Dieser Codeumsetzer, der unter einem den Teilnehmer identifizierenden Namen zur Verfügung gestellt wird, ist für die weitere Arbeit mit der Parallelcodierung die zentrale Grundlage. Er enthält alle mnemonischen Kurzwörter und ihnen beigeordnet die Informationen, die zur Erzeugung der maschineninternen Informationsdarstellung notwendig sind, sowie Strukturdaten, die das Codieren der Merkmale und das Decodieren, also die Rückfindung der Merkmale aus ihrer internen Verschlüsselung beschleunigen.

Wie immer beim Einsatz von Datenverarbeitungsanlagen gelten programmtechnische Beschränkungen. Ein Codeumsetzer kann die Daten für maximal 10 Ebenen aufnehmen. Definierte Ebenen werden dabei vom Teilnehmer mit Indizes versehen, die zwischen 1 und 10 liegen. Diese Ebenenindizes brauchen nicht fortlaufend angegeben zu werden, sondern müssen lediglich in den obigen Grenzen enthalten sein. Es können also beispielsweise die Ebenen 5, 7 und 1 erklärt werden, ohne daß überhaupt andere vorhanden sind.

Eine zweite Beschränkung begrenzt die Anzahl der von einem Codeumsetzer faßbaren Merkmale auf 300. Diese Festlegung wirkt sich nicht auf den Merkmalsumfang für die einzelnen Ebenen aus, solange nur für alle Ebenen zusammen nicht mehr als 300 Merkmale definiert werden. Es wäre also im Grenzfall möglich, eine Ebene allein mit 300 Merkmalen zu bestimmen.

Prinzipiell besteht zwischen Ebenen, die innerhalb eines Codeumsetzers angesprochen werden können, kein Zusammenhang. Ein Teilnehmer kann demnach, soweit es das Fassungsvermögen des Codeumsetzers zuläßt, die Daten aus verschiedenen Problembereichen, jeweils nach Ebenen aufgetrennt und mit voneinander verschiedenen Ebenenindizes versehen, in ein und denselben Codeumsetzer einbringen. Ein Zusammenhang wird erst durch den Teilnehmer selbst postuliert, etwa innerhalb eines Retrieval, wo mitgeteilt werden kann, auf welchen Ebenen zugleich angebbare Merkmale zutreffen müssen bzw. nicht zutreffen sollen.

Für sehr umfangreiche Unternehmungen kann der Fall eintreten, daß die Kapazität des Codeumsetzers nicht ausreicht. Dann sollte es möglich sein, eine Auftrennung nach verschiedenen disjunkten Problemkreisen, die unabhängig voneinander sind, vorzunehmen, so daß die Verwendung von mehreren Codeumsetzern schließlich zum Ziel führt. Allerdings besteht dann nur auf einer Anlage mit größerer Speicherkapazität die Möglichkeit, mit zwei Codeumsetzern simultan zu arbeiten. (Der Codeumsetzer ist für alle Programme kernspeicherresident. Eine nur in Teilen residente Aufspaltung könnte zwar die Gesamtkapazität beträchtlich steigern, allerdings auf Kosten der für die Ein/Ausgabeoperationen benötigten Zeit. Die Erfahrung wird erst zeigen, ob die bestehende Auslegung praktikabel ist.)

So wie nicht zwei Codeumsetzer zugleich verwendet werden können, besteht auch keine Beziehung zwischen den Codeumsetzern verschiedener Teilnehmer am System Parallelcodierung. Damit hat jeder Teilnehmer die Möglichkeit, mit seinem eigenen Begriffsspektrum zu arbeiten und unterliegt keinen Einschränkungen von außen.

Der dem Teilnehmer unter einem identifizierenden Namen zur Verfügung gestellte Codeumsetzer muß vor jeder weiteren Arbeit erzeugt werden. Als Vorbereitung hierzu ist notwendig die Aufspaltung nach Ebenen und die Vergabe von Ebenenindizes, sowie die Klassifizierung der Merkmalsvorräte für die einzelnen Ebenen. Dazu wird innerhalb einer Ebene den obligatorischen Merkmalen der Klassenindex 0 zugeordnet, und die Subklassen in der Menge der fakultativen Merkmale werden mit 1 beginnend fortlaufend numeriert. Damit wird jedes Merkmal in eine Ebene und innerhalb einer Ebene in eine Klasse verwiesen.

Insgesamt können auf einer Ebene höchstens 32 verschiedene Klassen angegeben werden, dann allerdings dürfte jede Klasse nur ein Element enthalten. Der andere Extremfall: ist nur eine Klasse für eine Ebene angegeben, so könnten theoretisch 2 Merkmale auf dieser Ebene verschlüsselt werden, wobei allerdings der Codeumsetzer nur 300 fassen kann. In der Regel wird ein konkretes Problem einen Zwischenweg beschreiben. Ist er im Sinne der Parallelcodierung nicht praktikabel, wird durch das den Codeumsetzer erstellende Programm die entsprechende Meldung gegeben.

Die den Ebenen und dort den einzelnen Klassen in der Form mnemonischer Kurzwörter zugeordneten Merkmale sind Eingabe für das Programm PC-10: ERSTELLEN EINES CODEUMSETZERS. Dieses Programm liefert als Ausgabe ein Protokoll der vereinbarten Merkmale, nach Ebenen und Klassen aufgetrennt, und druckt zu jedem Merkmal den in der Eingabe zur Erklärung angegebenen Kommentar. Darüberhinaus liefert es gegebenenfalls Fehlermeldungen, zusätzlich die im einzelnen vorgenommenen Codierungen, sowie eine Übersicht über den aufbereiteten Codeumsetzer. Das Protokoll der Merkmale dient als Arbeits-

unterlage für die weitere Codierung[6].

Die folgenden Abbildungen 1 bis 5 zeigen ein durch PC-10 erzeugtes Protokoll für drei Codierungsebenen, wie sie in der Forschungsstelle Freiburg des Instituts für deutsche Sprache in einem ersten Ansatz definiert wurden. Die in den Abbildungen abgedruckten Kommentare mögen an dieser Stelle zur Beschreibung der Merkmale genügen. Eine exakte Darstellung wird durch die Forschungsstelle zu gegebener Zeit erfolgen. An dieser Stelle soll lediglich ein Beispiel für eine mögliche Ebenenaufteilung, sowie eine Klassifikation innerhalb der Ebenen gegeben und eine Belegung der Merkmale mit mnemonischen Kurzwörtern gezeigt werden.

2.2 Updating eines Codeumsetzers

Im Rahmen fortschreitender Arbeiten kann sich nun die Notwendigkeit erweisen, neue Ebenen in den Codeumsetzer aufzunehmen oder Änderungen auf bereits festgelegten Ebenen vorzunehmen, dann nämlich, wenn sich bisherige Definitionen als zumindest teilweise unzulänglich erwiesen, oder wenn ein Problembereich erweitert werden soll.

Falls in den Codeumsetzer nur die Merkmale zu neuen Ebenen aufgenommen werden sollen, steht das Programm PC-11: ERWEITERN EINES CODEUMSETZERS zur Verfügung. Eingabe für dieses Programm sind die neu festgelegten Merkmale, gruppiert nach Ebenen (deren Indizierung sich nicht mit der Indizierung bereits vorhandener Ebenen deckt), und innerhalb der Ebenen wie bereits bekannt klassifiziert. Ein Umcodieren schon bestehender Daten ist nicht notwendig, da diese keine Referenz zu den neu eingebrachten Ebenen

CODIERUNGSEBENE 1 WORTARTENEBENE

KLASSE	MNEMONISCHES KURZWORT	BEMERKUNGEN
0	....	U N C O D I E R T
0	VRB	FINITES VERB, UNVERKETTET
0	VRB1	FINITES VERB, VERKETTET MIT INFINITIV
0	VRB2	FINITES VERB, VERKETTET MIT PARTIZIP II
0	VRB3	FINITES VERB, VERKETTET MIT INFINITIV UND PARTIZIP II
0	VZS	VERBZUSATZ
0	PTZ1	PARTIZIP I
0	PTZ2	PARTIZIP II
0	INF	INFINITIV
0	INFZ	INFINITIV MIT 'ZU'
0	SUB	SUBSTANTIV
0	ART	ARTIKEL UND DEMONSTRATIVA
0	PERS	PERSONALPRONOMEN
0	REFL	REFLEXIVPRONOMEN
0	POSS	POSSESIVPRONOMEN
0	FRPR	FRAGEPRONOMEN
0	PRON	RESTGRUPPE DER PRONOMINA (QUANTOREN)
0	ADJ1	ADJEKTIV, DAS ADNOMIAL STEHEN KANN
0	ADJ2	ADJEKTIV, DAS ADNOMIAL STEHT
0	ADV	ADVERB
0	ADVN	ADVERB 'NICHT'
0	KONK	KONJUNKTION, KOORDINIEREND
0	KONS	KONJUNKTION, SUBORDINIEREND
0	PREP	PRAEPOSITION/POSTPOSITION
0	ANTP	ANTWORTPARTIKEL
0	ZERO	UNFLEKTIERTES ELEMENT, DESSEN WORTART NICHT ENTSCHEIDBAR IST
0	SPRW	SPRECHERWECHSEL
0	FR	FRAGE [?]
0	E	EMPHASE [:]
0	P	SPRECHPAUSE [+p+]
0	Z	ZITATE ODER EIGENNAMEN [z+ +z]
0	F	FREMDSPRACHE UND DIALEKTE [f+ +f]
0	B	BINDESTRICH [-]
0	K	SIMULTANES SPRECHEN [k+ +k]

Abbildung 1

CODIERUNGSEBENE 1 (FORTSETZUNG)

1	LE	LEERES ELEMENT	WORTEINHEITEN IN /-SAETZEN
1	GRF	GRAMMATISCHER FEHLER	
2	STV	STARKES VERB	MORPHOLOGISCHE INFORMATION
2	SWV	SCHWACHES VERB	
2	STVS	VERB 'SEIN'	
2	FV	FUNKTIONSVERB	
3	KOMP	KOMPERATIV	STEIGERUNG DES ADJEKTIVS
3	SUP	SUPERLATIV	
4	MOD	MODAL	INFORM. ZUR KONJUNKTION
4	TEMP	TEMPORAL	
4	KAUS	KAUSAL	
4	FIN	FINAL	

Abbildung 2

CODIERUNGSEBENE 2 WORTGRUPPENEBENE

KLASSE	MNEMONISCHES KURZWORT	BEMERKUNGEN	
0	....	U N C O D I E R T	
0	VG	VERBALGRUPPE	
0	NG	NOMINALGRUPPE	
1	1P	1.PERSON	INFORM. ZUR PERSON
1	2P	2.PERSON	
1	3P	3.PERSON	
1	ANR	ANREDE	
2	SI	SINGULAR	NUMERUSINFORMATION
2	PL	PLURAL	
3	PRE	PRAESENS	TEMPUSINFORMATION
3	IPF	IMPERFEKT	
3	PER	PERFEKT	
3	PQU	PLUSQUAMPERFEKT	
3	F1	FUTUR I	
3	F2	FUTUR II	
4	AK	AKTIV	GENUSINFORMATION
4	WP	WERDEN - PASSIV	
4	SP	SEIN - PASSIV	

Abbildung 3

CODIERUNGSEBENE 2 (FORTSETZUNG)

5	ID	INDIKATIV	MODUSINFORMATION
5	KJ	KONJUNKTIV	
5	KJW	KONJUNKTIV 'WUERDE'	
5	IMP	IMPERATIV	
6	NOM	NOMINATIV	KASUSINFORMATION
6	GEN	GENITIV	
6	DAT	DATIV	
6	AKK	AKKUSATIV	
6	PP	PRAEPOSITIONALPHRASE	
6	IK	IDENTIFIKATIONSKASUS	
7	MOD	MODAL	ZUSATZINFORMATION ZUR NG
7	TEMP	TEMPORAL	
7	KALZ	KALENDERZEIT / UHRZEIT	
7	TEKA	TEMPORAL / KAUSAL	
7	KAUS	KAUSAL	
7	LOK	LOKAL	
7	LEER	FUER DIE POSITION 'ES'	
8	ATT	ATTRIBUTIVE NOMINALGRUPPE	STATUSINFORMATION

Abbildung 4

CODIERUNGSEBENE 3 — SATZEBENE

KLASSE	MNEMONISCHES KURZWORT	BEMERKUNGEN	
0	....	U N C O D I E R T	
0	HS	HAUPTSATZ	
0	EHS	EINGESCHLOSSENER HAUPTSATZ [.+ +.]	
0	AH	ABHAENGIGER HAUPTSATZ [s+ +s]	
0	NS	NEBENSATZ [,+ +,]	
0	IS	INFINITIV [i+ +i]	
0	DV	DISKONTINUIERLICHE VERSCHACHTELUNG [n+ +n]	
0	SS	SCHRAEGSTRICHSATZ	
0	KS	KURZSATZ	
0	PT	PARENTHESE [( )]	
1	EST	VERBENDSTELLUNG	VERBSTELLUNG IM NEBENSATZ
1	NEST	KEINE VERBENDSTELLUNG	
2	SELE	SEMANTISCH LEERES TEMPUS	TEMPORALE RELATION
2	AUES	AKTZEIT UEBERLAPPT SPRECHZEIT	
2	AVS	AKTZEIT VOR SPRECHZEIT	
2	ANS	AKTZEIT NACH SPRECHZEIT	
2	AVK	AKTZEIT VOR KONTEXTZEIT	

Abbildung 5

haben. Die Erweiterung eines Codeumsetzers um neue Ebenen ist so lange möglich, wie die Gesamtanzahl aller im Codeumsetzer enthaltenen Merkmale 300 nicht überschreitet und nicht mehr als 10 Ebenen definiert werden.

Es ist zum anderen aber auch die Erweiterung des Merkmalsvorrats bereits bestehender Ebenen um neue Merkmale, sowie das Herausnehmen von Merkmalen möglich. Generell kann man sich dabei so verhalten, als gäbe es noch keinen Codeumsetzer, das heißt: eine neue Codierung wird, im wesentlichen unabhängig von der bestehenden, festgelegt. Da aus ihr in der Regel eine andere interne Verschlüsselung resultiert, ist es unumgänglich, daß bestehende Datenmengen umcodiert werden, da diese nur durch den alten Codeumsetzer interpretiert werden können.

Man kann davon ausgehen, daß die Ebenenindizierung erhalten bleibt, insbesondere auch für die Ebenen, in denen eine Veränderung am Merkmalsvorrat vorgenommen werden soll. Schließlich ist die Ebenenindexvergabe nur eine identifizierende Maßnahme, die keine Präferenzordnung nach sich zieht. Mit dieser Reglementierung wird jetzt ein neuer Codeumsetzer definiert. Dabei müssen Merkmale, die bereits im alten Codeumsetzer enthalten waren, mit den gleichen mnemonischen Kurzwörtern belegt werden. Neue Merkmale können hinzugenommen werden, darüberhinaus brauchen nicht alle Merkmale übernommen zu werden. Ebenen, die unverändert bleiben, müssen dennoch neu mit eingegeben werden, wozu man die bereits vorliegenden Unterlagen verwenden kann. Ohne erst später auf PC-11 zurückzugreifen, können Merkmale zu neuen Ebenen bereits an dieser Stelle eingebracht werden.

Die so schließlich definierten Merkmale, bzw.

die ihnen zugeordneten mnemonischen Kurzwörter mit ihren Ebenen- und Klassenindizes, werden durch das Programm PC-12: VERÄNDERN EINES CODEUMSETZERS verarbeitet. Dieses Programm legt unter dem den Teilnehmer identifizierenden Namen den neuen Codeumsetzer ab und reserviert den alten in einer Systemdatei solange, bis alle notwendigen Umcodierungen abgeschlossen sind.

Sämtliche bestehenden Codierungen, die unter dem alten Codeumsetzer erzeugt worden waren, müssen nun dem neuen Codeumsetzer zugänglich gemacht werden[7]. Diese Aufgabe übernimmt das Programm PC-60: UMCODIEREN FÜR NEUEN CODEUMSETZER. Da ein Umcodieren nur notwendig ist für Ebenen, in deren Merkmalsvorrat verändernd eingegriffen wurde, werden dem Programm diese Ebenen mitgeteilt. Mit Hilfe des alten Codeumsetzers wird dann die bestehende Codierung entschlüsselt und die resultierenden mnemonischen Kurzwörter werden im neuen Codeumsetzer aufgesucht. Ist eines auf der gleichen Ebene nicht mehr vorhanden, wird angenommen, daß darauf nicht mehr Bezug genommen wird, die Information wird also gelöscht. Andernfalls wird schließlich mit dem neuen Codeumsetzer die neue interne Verschlüsselung erzeugt, so daß nach Abschluß der Umcodierung ausschließlich mit dem neuen Codeumsetzer gearbeitet werden kann. Es ist klar, daß Merkmale, die neu definiert wurden, dabei nicht erzeugt werden können, da sie kein Äquivalent im alten Codeumsetzer haben. Allerdings kann nun ein Datenbestand um solche Merkmale erweitert werden.

3. CODIEREN DURCH PARALLELCODIERUNG

3.1 Das textliche Ausgangsmaterial

Das Objekt einer Analyse, die unterstützt durch die Parallelcodierung durchgeführt werden soll, sind Texte gleich welcher Sprache. Diese Texte müssen in maschinenverarbeitbarer Form vorliegen, üblicherweise auf Magnetbändern. Bei der Parallelcodierung sollte von einer endgültigen Fassung der Texte ausgegangen werden. Eine nachträgliche Änderung des Textmaterials wird in der Regel seine Struktur derart verändern, daß der in der zugehörigen Codierung enthaltene Bezug zum Text verloren geht oder zumindest verfälscht wird.

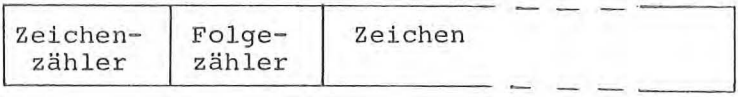
Ein Text ist in Einheiten (Records) zerlegt und als Folge solcher Einheiten abgespeichert. Die Segmentierung kann nach beliebigen formalen Gesichtspunkten erfolgen. So sind beispielsweise die Texte des Mannheimer Korpus segmentiert in syntaktische Sätze, ebenso auch die Texte zur gesprochenen Sprache in Freiburg. Denkbar wäre aber auch eine Zerlegung nach anderen Gesichtspunkten, etwa in Paragraphen bei Gesetzestexten.

Ein so festgelegter Record ist die größte Texteinheit, die mit der Parallelcodierung erfaßt werden kann. Diese Einheit darf für die bestehende Version des Systems höchstens 5000 Zeichen umfassen.

Leerzeichen, die in die Zeichenfolge des Records eingebettet sind, zerlegen den Inhalt der

Texteinheit in Wörter, jedenfalls im Sinne der Parallelcodierung. Über die Wörter kann schließlich der Bezug einer Codierung zu einem Text hergestellt werden, es können aber auch Zeichenketten angegeben werden, die nicht als Wörter erscheinen.

Jeder Record enthält die Information über seine Länge (d. h. die Anzahl der Zeichen, die in das Segment eingelagert sind), sowie eine Folgenummer, die mit einem beliebigen, nicht negativen Startwert für den ersten Record die Folge der Records fortlaufend in Einerschritten numeriert.



Ein derart aufgebauter Record, im allgemeinen von variabler Länge, kann als Textgrundlage zur Parallelcodierung dienen.

Der erste Record eines jeden Textes enthält nur drei Zeichen, die den Textschlüssel bilden und die Verwaltung der Texte vereinfachen. Auf einem Magnetband können mehrere Texte, durch Dateiendemarken getrennt, enthalten sein.

3.2 Erstellen von Textvorlagen zur Codierung

Codieren im Sinne der Parallelcodierung heißt nun, eine Textsequenz und die dafür zutreffen-

den Merkmale angeben. Das ist eine ordnende, manuelle Tätigkeit, die man allgemein aufspalten wird in

- skizzenhafte Niederschrift durch den analysierenden Linguisten
- Übertragen in Ablochschemata
- Abschrift auf Lochkarten oder andere Datenträger.

Das System bietet gewisse Hilfen, welche die Ausführung dieser Tätigkeiten unterstützen und teilweise überflüssig machen.

1) Aufbereitung eines Textes als Ablochunterlage

Die Programme PC-20: ABLOCHUNTERLAGEN VON SATZZERLEGTEM TEXT und PC-21: ABLOCHUNTERLAGEN VON SATZZERLEGTEM TEXT (FKZ-BAND) erzeugen einen formularmäßigen Ausdruck (siehe Abbildung 6), in den der Linguist seine Codierungen auf bis zu vier Ebenen gleichzeitig niederschreiben kann, und aus dem unmittelbar abgelocht werden kann, ohne daß die Daten erst auf ein Ablochschema übertragen werden müssen. (Eine ausführliche Beschreibung der Handhabung dieses Formulars, sowie die Anweisungen zum Ablochen daraus befinden sich in Anhang I.)

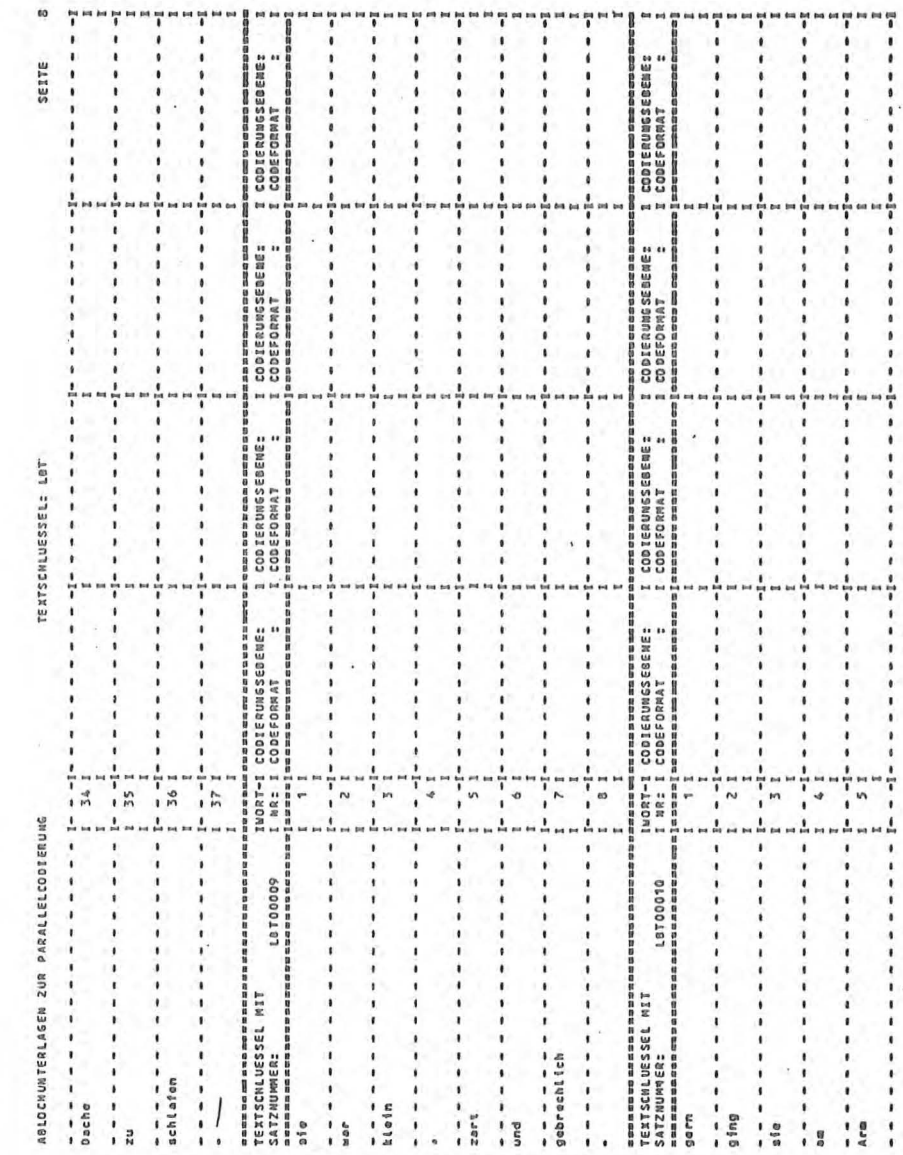
Bei diesem Ausdruck werden die einzelnen Records eines Textes, durch Kopfzeilen voneinander getrennt, so aufgelistet, daß für jedes Wort des Records eine Zeile reserviert wird. Zugleich werden die Wörter fortlaufend numeriert - diese Numerierung unterstützt die Zuweisung der Textsequenzen -, und schließlich stellt die Spalteneinteilung des Formulars auf bis zu vier Ebe-

nen für jedes Wort ein Kästchen bereit, in das die handschriftlichen Analyseergebnisse eingetragen werden. Wo keine Codierung zutrifft, bleibt dieses Kästchen leer. Es wird ein Kästchen ausgefüllt, wenn eine Codierung sich auf mehrere Wörter zugleich bezieht.

In dem Programm PC-20, das auf umfangreichere Texte angewendet wird, die als einzige Datei auf einem Magnetband liegen, ist auch die Möglichkeit vorgesehen, daß nur Teile des Textes formularmäßig aufbereitet werden. Welche Teile gedruckt werden, kann durch eingrenzende Folgenummern von Records angegeben werden.

Das Programm PC-21 dagegen legt ein Textband zugrunde, das mehrere als Files getrennte Texte in nicht notwendig bekannter Reihenfolge enthält. Über Anfrage an den Operator wird dabei die Textauswahl getroffen. Jeder aufzubereitende Text wird hierbei vollständig abgedruckt.

Abbildung 6


ABLOCKUNTERLAGEN ZUR PARALLELCODIERUNG
TEXTSCHLUESSEL: LBT
SEITE 8
Dache
zu
schlafen
TEXTSCHLUESSEL MIT SATZNUMMER: LBT00009
sie
war
klein
,
zart
und
gebrechlich
.
TEXTSCHLUESSEL MIT SATZNUMMER: LBT00010
garn
ging
sie
es
Arm
IWORT-NR:
CODIERUNGSEBENE:
CODEFORMAT

```
WORTNUMERIERTE TEXTVORLAGE ZUR PARALLELCODIERUNG          TEXTSCHLUESSEL: xaa          SEITE   8
```

```
xaa00049    (SPALTEN 1 BIS 11)
----------

und ( wie gesagt ) diese Schwerter sind ja auch47 etwas ungemütlich27 ,+ wenn man ihnen zu4 nahe kommt09 +, .
01  0203  04     0506    07        08   09 10     11    12            13 14   15  16    17  18   19       20 21

xaa00050    (SPALTEN 1 BIS 11)
----------

man läßt also den Fisch4 sich austoben26 versucht26 ,+ sobald er seine5 Kreise4 zieht36 etwas näher an s Boot kommt26 +, i+ die
01  02   03   04  05     06   07         08          09 10     11 12     13      14      15    16    17 1819   20       21 22 23

Leine weiter hineinzunehmen26 +i i+ um ihn dadurch langsam an die Oberfläche zu bekommen16 +i .
24    25     26               27 28 29 30  31      32      33 34  35         36 37         38 39

xaa00051    (SPALTEN 1 BIS 11)
----------

,+ und wenn er dann ziemlich nahe an der Oberfläche ist46 und schon etwas ermüdet ist26 +, versucht man57 i+ ihm eine Schlinge47
01 02  03   04 05   06       07   08 09  10          11    12  13    14    15      16    17 18       19    20 21  22   23

über den Schwanz zu ziehen26 +i i+ und4 mit Hilfe einer Winde4 den Fisch4 herauszuholen09 +i .
24   25  26      27 28       29 30 31   32  33    34    35     36  37    38              39 40

xaa00052    (SPALTEN 1 BIS 11)
----------

das4 is ziemlich schwierig26 .
01   02 03       04          05

xaa00053    (SPALTEN 1 BIS 11)
----------

da müssen also alle Mann ran26 i+ um den Fisch rüberzuholen26 +i .
01 02     03   04   05   06    07 08 09  10    11             12 13

xaa00054    (SPALTEN 1 BIS 11)
----------

ich war aber heilfroh26 ,+ als wir nun ersten an Bord haben49 +, .
01  02  03   04        05 06  07  08  09    10 11   12      13 14
```

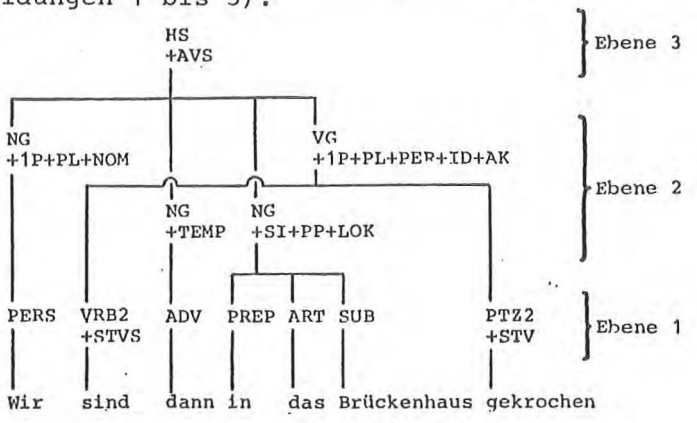
Abbildung 7

2) Wortnumerierte Ausgabe von Texten

Wird nur auf einer einzigen Ebene codiert, ist die oben beschriebene Formularaufbereitung zu aufwendig. Für diesen Fall kann durch das Programm PC-22: WORTNUMERIERTE TEXTAUSGABE eine andere Form der Textaufbereitung zur Parallelcodierung gewählt werden. Dabei wird jeder Record zeilenweise ausgedruckt, die einzelnen Wörter sind durch darunterstehende Indizes ebenfalls fortlaufend numeriert (siehe Abbildung 7). Hierbei sollte allerdings vom bearbeitenden Linguisten direkt in ein Ablochschema codiert werden, da diese Form der Textaufbereitung keine so übersichtliche Codierung zuläßt, daß die Fehlerrate beim Abschreiben der Daten klein gehalten werden könnte. Im übrigen gelten auch hier die Codierungsanweisungen im Anhang I.

3.3 Codieren vorbereiteter Texte

Ein Beispiel für eine mögliche Codierung eines Satzes ist die folgende Strukturdarstellung. Verwendet sind dabei die mnemonischen Kurzwörter der Freiburger Forschungsstelle (siehe Abbildungen 1 bis 5).



Diese Darstellung gibt Beispiele von Eigenschaften und zeigt ihre Zuordnung zum Text. Die Notation von Deskriptoren in dieser Form ist, wenn die mnemonischen Kurzwörter geschickt gewählt sind, unmittelbar lesbar.

An dieser Stelle sollen nur die unter dem Analysemodell zu konkretisierenden Eigenschaften behandelt werden. Sie werden formuliert als Folge mnemonischer Kurzwörter, die auf den zugehörigen Ebenen erklärt und im Codeumsetzer enthalten sind. In jeder Folge ist höchstens ein obligatorisches Merkmal enthalten. Fehlt es und sind zugleich fakultative Merkmale enthalten, wird vom Verarbeitungsprogramm selbstständig das Kurzwort "...." substituiert, und so erscheint es in Zukunft immer, wenn die zugehörige Codierung in ihrer Ursprungsform erzeugt wird, es sei denn, es wird korrigierend überschrieben. An das obligatorische Merkmal reihen sich nun zutreffende fakultative Merkmale an, die geeignet sind, den vorliegenden Sachverhalt in Einzelheiten zu beschreiben. Inwieweit sie angegeben werden, ist vom Benutzer selbst und seinen Zielsetzungen abhängig. Generell aber sollte man möglichst viel Information durch einen Arbeitsgang allein einbringen, auch wenn ihre Auswertung erst zu einem späteren Zeitpunkt vorgesehen ist.

Die einzelnen Merkmale selbst sind voneinander unabhängig, das heißt: sie können in beliebiger Reihenfolge angegeben werden und sie stammen aus disjunkten Klassen. Das zweite Kriterium ergibt sich aus der Konstruktion des Codeumsetzers. Das Verarbeitungsprogramm prüft lediglich, ob nicht zwei obligatorische Merkmale zugleich angegeben sind und meldet gegebenenfalls einen Fehler. Eine Prüfung auf Unabhängigkeit der fakultativen Merkmale erfolgt nicht. Werden demnach zwei Merkmale einer Klasse zugleich angegeben, resultiert eine nicht kontrollierbare interne Ver-

schlüsselung, aus der in der Regel nicht mehr alle einzelnen Merkmale ableitbar sind.

Eine programmtechnische Beschränkung gilt auch hier: es dürfen höchstens 10 Merkmale innerhalb einer Codierung angegeben werden.

3.4 Codierung und Textbezug

Höchste Einheit im Sinne der Parallelcodierung ist die vollständige Sequenz, die in einem Record enthalten ist und die wir im folgenden als Satz bezeichnen. Das bedeutet, ein Deskriptor als Folge mnemonischer Kurzwörter kann höchstens auf einen Satz zugreifen und nicht auf mehrere zugleich.

Durch ein Textaufbereitungsprogramm wird dieser Satz als numerierte Wortfolge wiedergegeben. Für das Beispiel von Seite 45 hätte das Programm PC-22 folgende Ausgabe:

Wir sind dann in das Brückenhaus gekrochen .
01 02 03 04 05 06 07 08

(Die den Textwörtern zugeordneten Nummern der zweiten Zeile werden im folgenden als Wortindizes bezeichnet.)

Wir gehen zuerst davon aus, daß die Textsequenzen, auf die sich unsere Codierungen beziehen, Wörter oder Wortfolgen sind und nicht nur Teile von Wörtern. Prinzipiell lassen sich hierbei vier Fälle unterscheiden, nämlich:

zutreffende Sequenz ist

- genau ein Wort
- eine zusammenhängende Folge von Wörtern
- eine nicht zusammenhängende Folge von Wörtern
- der ganze Satz.

Zuordnungen von Codierungen zum Text lassen sich dann als P a a r e von Wortindizes angeben, wobei jedes Paar das erste und letzte Wort der Sequenz angibt. Ein Wort, sowie eine zusammenhängende Folge von Wörtern oder der ganze Satz könnte dann durch ein solches Paar beschrieben werden, eine nicht zusammenhängende Folge von Wörtern dagegen durch so viele Paare, wie die Sequenz zusammenhängende Teile hat.

Sequenzen aus dem obigen Beispielsatz könnten dann folgendermaßen beschrieben werden:

dann	3,3
in das Brückenhaus	4,6
der ganze Satz	1,8
wir sind gekrochen	1,2,7,7

Das System läßt nun Vereinfachungen zu, ohne dabei die obigen Formen der Zuordnung als fehlerhaft zu monieren.

1) Die Sequenz ist genau ein Wort

Es genügt, den Index des Wortes allein anzugeben, etwa für

dann 3

2) Sequenz ist der ganze Satz

Es brauchen in diesem Fall gar keine zuordnenden Angaben gemacht zu werden.

In den beiden übrigen Fällen müssen die Zuordnungen immer paarig sein, auch wenn für eine nicht zusammenhängende Folge von Wörtern ein Glied der Folge aus nur einem Wort besteht. Die Angaben 1,2,7 für die Sequenz *wir sind gekrochen* wäre also fehlerhaft.

Die zweite Möglichkeit des Bezugs einer Codierung zum Text liegt in der Abgrenzung von Wortbestandteilen, also von Zeichenketten. Diese Form der Zuordnung wird dann gewählt, wenn die codierte Textsequenz nicht als ganzes Wort dargestellt ist. Die Angaben für Zeichenketten bestehen prinzipiell aus Z a h l e n t r i p e l n in der allgemeinen Form

$$w,z_1,z_2$$

Dabei bedeutet w den Index des Wortes, in dem die Zeichenkette beginnt, z_1 gibt das erste Zeichen und z_2 das letzte Zeichen der zutreffenden Zeichenkette an, jeweils gezählt vom Anfang des durch w bezeichneten Wortes. Eine Sequenz, die aus nicht zusammenhängenden Zeichenketten besteht, wird dann beschrieben als Folge solcher Zahlentripel, wobei jedes Glied wieder eine zusammenhängende Zeichenkette beschreibt. Nehmen wir wieder Bezug auf den Beispielsatz, dann wird bezeichnet:

ge	7,1,2
Haus	6,8,11

Brücke	6,1,6
wir krochen	1,1,4,7,3,9

(Wäre im letzten Fall 1,1,3,7,3,9 geschrieben, würde das trennende Leerzeichen entfallen und wir erhielten: *wirkrochen*.)

Diese beiden Fälle, Bezug auf Wörter oder Bezug auf Zeichenketten, müssen voneinander unterschieden werden. Dazu wird ein sogenanntes C o d e - f o r m a t eingeführt, das die Werte hat

Codeformat = 0 für Wortketten

Codeformat = 1 für Zeichenketten

Die zuordnenden Zahlenangaben heißen in der Parallelcodierung Positionsangaben. Die programmtechnischen Beschränkungen, die für Positionsangaben gelten, sind

höchstens 6 Paare für Codeformat 0

höchstens 4 Tripel für Codeformat 1

Vom Verarbeitungsprogramm werden die Positionsangaben prinzipiell auf Zeichenkettenbegrenzung gesetzt, wobei Beginn und Ende jeder Zeichenkette vom Satzanfang errechnet wird.

Die für eine Textsequenz fixierte Folge mnemonischer Kurzwörter, im folgenden Codierungsangaben genannt, verbunden mit den zugehörigen Positionsangaben, bilden dann eine eindeutige Codierung, wie sie auch durch die Darstellung im Strukturbaum erreicht wird.

Um nun die Fülle der Codierungen zu einem Text zu ordnen und den Bezug zum ganzen Text zu sichern, treten zu einer vollständigen Codierung

noch weitere Angaben. Einmal hilft das Codeformat, die Positionsangaben zu interpretieren. Durch die Angabe des zutreffenden Ebenenindex für die Codierungsangaben werden auch diese erklärt. Schließlich wird noch der Bezug der Codierung zum Satz selbst durch die Angabe der Satznummer gesichert. Der Vollständigkeit halber, allerdings nur für die manuelle Datenverwaltung, tritt der Textschlüssel noch hinzu. Somit besteht eine vollständige Codierung aus

a: Textschlüssel

b: Satznummer

c: Ebenenindex

d: Codeformat

e: Codierungsangaben

f: Positionsangaben

Beim Ablochen lassen sich

- innerhalb einer Codierungsebene die Angaben a bis d
- innerhalb eines Satzes die Angaben a bis b
- innerhalb eines Textes die Angabe a

automatisch duplizieren, so daß im wesentlichen imeer nur die Codierungs- und Positionsangaben geschrieben werden müssen. Diese Angaben können unformatiert und spaltenfrei in die ihnen zugewiesenen Felder einer Lochkarte übertragen werden (siehe dazu Anhang I). Sie bilden die vollständige Codierung eines Einzelphänomens.

Das Codeformat hat keine globale Gültigkeit, sondern kann von Codierung zu Codierung wechseln. Für die Form der Zuordnung durch Positionsangaben kann also die für den Einzelfall zweckmäßigste Darstellung gewählt werden.

Abbildung 8

TEXTSCHLUESSEL	SATZNUMMER	EBENENINDEX	CODEFORMAT	CODIERUNGS-ANGABEN	POSITIONS-ANGABEN
TXT	00386	01	0	PERS	1
TXT	00386	01	0	VRB2+STVS	2
TXT	00386	01	0	ADV	3
TXT	00386	01	0	PREP	4
TXT	00386	01	0	ART	5
TXT	00386	01	0	SUB	6
TXT	00386	01	0	PTZ2+STV	7
TXT	00386	02	0	NG+1P+PL+NOM	1
TXT	00386	02	0	VG+1P+PL+PER+ID+AK	2,2,7,7
TXT	00386	02	0	NG+TEMP	3
TXT	00386	02	0	NG+SI+PP+LOK	4,6
TXT	00386	03	0	HS+AVS	

Die Codierung des Beispielsatzes, wie sie aus der Strukturbaum-Darstellung entnommen werden kann (Seite 45), erfolgt dann schließlich in der Form, wie sie aus Abbildung 8 ersichtlich ist. Textschlüssel ist dabei TXT, der Beispielsatz sei der 386-te dieses Textes.

3.5 Datenaufnahme und Fehlerkorrektur

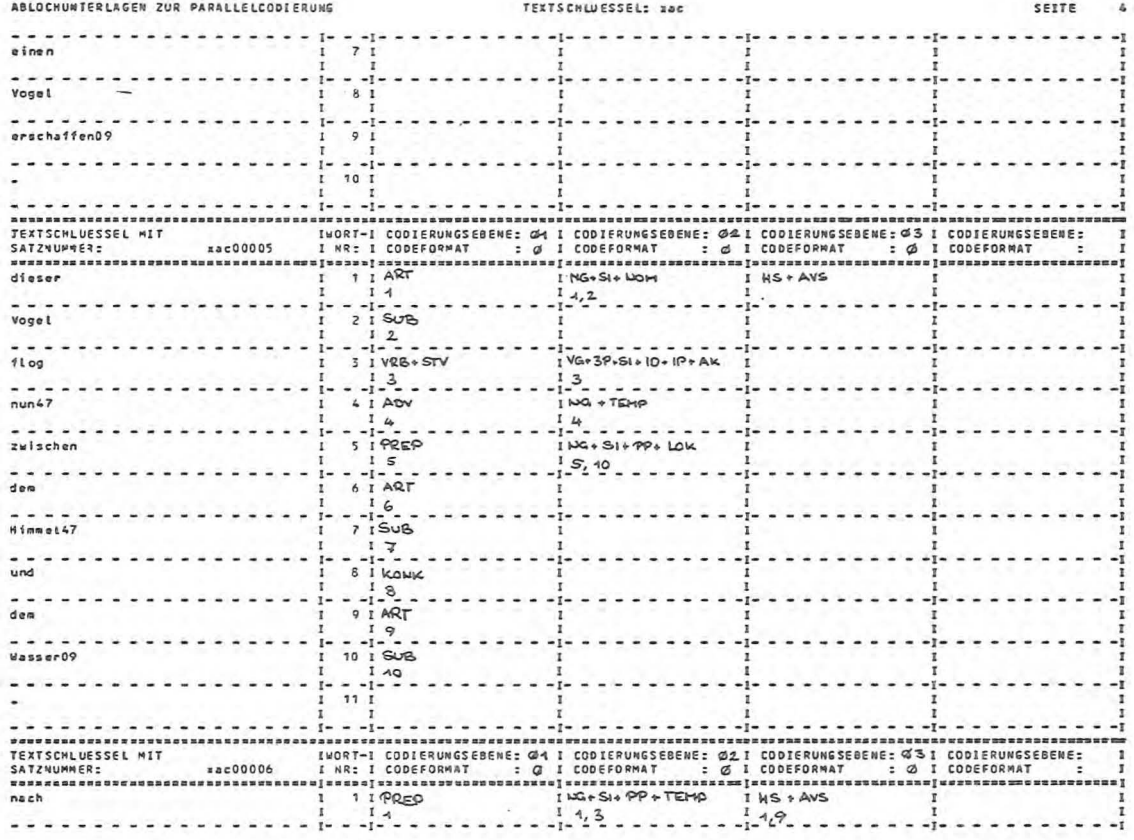
Die Abbildungen 9 und 10 zeigen eine Codierung, wie sie mit Hilfe eines Textausdruckes durch PC-20 oder PC-21 handschriftlich erstellt werden kann. Die darin enthaltenen Angaben werden abgelocht und sind dann Eingabe für das Programm PC-30: KONTROLLE UND UMWANDLUNG VON CODIERUNGEN.

Es ist klar, daß das Ablochen eine Fehlerquelle bedeutet, welche die Daten verfälschen oder unkenntlich machen kann. Fehler können aber auch beim handschriftlichen Eintragen der Daten in die Codierblätter auftreten. Demnach müssen wir zwei Arten von Fehlern unterscheiden, nämlich formale, die das Programm außerstande setzen, die Daten zu interpretieren, und inhaltliche, wobei zwar Ergebnisse erhalten werden können, aber diese Ergebnisse einen nicht zutreffenden Sachverhalt kennzeichnen.

Formale Fehler in den Daten können vom Programm PC-30 erkannt werden. Für mögliche Fehler ist ein kennzeichnender Fehlercode vorgesehen, der durch das Programm in der Form

**** n **** Inhalt der fehlerhaften Lochkarte *****

erzeugt wird und in das Ausgabeprotokoll des Pro-



ABLOCHUNTERLAGEN ZUR PARALLELCODIERUNG TEXTSCHLUESSEL: zac SEITE 4

einen	7				
Vogel	8				
erschaffen09	9				
.	10				

TEXTSCHLUESSEL MIT SATZNUMMER: zac00005	WORT-NR:	CODIERUNGSEBENE: Ø1 CODEFORMAT : Ø	CODIERUNGSEBENE: Ø2 CODEFORMAT : Ø	CODIERUNGSEBENE: Ø3 CODEFORMAT : Ø	CODIERUNGSEBENE: CODEFORMAT :
dieser	1	ART 1	NG+SI+NOM 1,2	HS+AVS	
Vogel	2	SUB 2			
flog	3	VRB+STV 3	VG+3P+SI+ID+IP+AK 3		
nun47	4	ADV 4	NG+TEMP 4		
zwischen	5	PREP 5	NG+SI+PP+LOK 5,10		
dem	6	ART 6			
Himmel47	7	SUB 7			
und	8	KONK 8			
dem	9	ART 9			
Wasser09	10	SUB 10			
.	11				

TEXTSCHLUESSEL MIT SATZNUMMER: zac00006	WORT-NR:	CODIERUNGSEBENE: Ø1 CODEFORMAT : Ø	CODIERUNGSEBENE: Ø2 CODEFORMAT : Ø	CODIERUNGSEBENE: Ø3 CODEFORMAT : Ø	CODIERUNGSEBENE: CODEFORMAT :
nach	1	PREP 1	NG+SI+PP+TEMP 1,3	HS+AVS 1,9	

einigem	2	PROJ 2			
flug	3	SUB 3			
ermüdete	4	VRB + SWV 4	VG+3P+S1+ID+IPF+AKK 4		
er56	5	PERS 5	NG+3P+S1+NOM 5		
und4	6	KONK 6			
suchte	7	VRB + SWV 7	VG+3P+S1+ID+IPF+AKK 7		
ein	8	ART 8	NG+S1+AKK 8,9		
plätzchen47	9	SUB 9			
i+	10			IS 10,16	
um	11	KONS 11			
sich	12	REFL 12	VG 12,15		
niederlassen	13	INF + STV 13			
zu	14	INFZ 14			
können09	15	INFZ + SWV 15			
+i	16				
.	17				

TEXTSCHLUESSEL MIT SATZNUMMER: xac00007	WORT-NR:	CODIERUNGSEBENE: CODEFORMAT :	CODIERUNGSEBENE: CODEFORMAT :	CODIERUNGSEBENE: CODEFORMAT :	CODIERUNGSEBENE: CODEFORMAT :
er	1				
fand	2				

Abbildung 10

gramms dort eingefügt wird, wo sich die fehlerhafte Karte befindet. Der Inhalt der Lochkarte wird dabei durch das Zeichen ":" in die Felder

Textschlüssel
Satznummer
Ebenenindex
Codeformat
Codierungsangaben
Positionsangaben

aufgeteilt. Die möglichen Werte von n mit den zugeordneten Fehlerquellen sind aus Abbildung 11 ersichtlich.

Karten, deren Inhalt mit einem dieser Kennzeichen aufgelistet sind, wurden abgewiesen, die in ihnen enthaltenen Informationen müssen in einem folgenden Korrekturlauf neu eingegeben werden.

Alle Daten, die keine formalen Fehler haben, werden durch das Programm verschlüsselt und abgespeichert. Zugleich wird ein Ausgabeprotokoll gefertigt, das die Codierungsangaben, jetzt formatiert, abdruckt und den Codierungsangaben gleich die zugeordnete Textsequenz (und nicht nur die Positionsangaben) gegenüberstellt. In diesem Protokoll sind auch die fehlerhaften Karten enthalten.

Verweisen die Eingabedaten auf einen neuen Satz, wird dieser Satz vollständig abgedruckt. Dann erfolgt die Ausgabe der Codierungen zum Satz mit den zugehörigen Textsequenzen, und zwar in der Reihenfolge des Eingangs (siehe Abbildungen 12 und 13). Alle formal als richtig erkannten Codierungen werden dabei fortlaufend numeriert.

Abbildung 11

KENNUNG FORMALER FEHLER BEI DER DATENAUFNAHME ZUR PARALLEL-CODIERUNG

**** 0 ****	IN DEN FELDERN FUER SATZNUMMER, CODIERUNGSEBENE ODER CODEFORMAT IST EINE LOCHUNG ENTHALTEN, DIE KEIN ZIFFERNZEICHEN BEDEUTET. (ALLE ZEICHEN DIESER FELDER WURDEN AUF 0 GESETZT.)
**** 1 ****	IN DEN CODIERUNGSANGABEN FEHLT EIN TRENNENDES + ODER EIN MNEMONISCHES KURZWORT IST LAENGER ALS VIER ZEICHEN.
**** 2 ****	ZWEI ZAHLEN DER POSITIONSANGABEN WURDEN OHNE TRENNENDES KOMMA GESCHRIEBEN ODER ES WURDE EIN ZEICHEN IM POSITIONSFELD GELOCHT, DAS KEIN ZIFFERNZEICHEN ODER KOMMA IST.
**** 3 ****	ES WURDE EINE SATZNUMMER ANGEGEBEN, DIE UEBER DEN VORHANDENEN TEXT HINAUS VERWEIST.
**** 4 ****	DIE CODIERUNGSANGABEN ENTHALTEN EIN MNEMONISCHES KURZWORT, DAS AUF DER ANGEGEBENEN EBENE NICHT ERKLAERT IST ODER DIE EBENE SELBST IST NICHT ERKLÄRT.
**** 5 ****	DIE POSITIONSANGABEN UNTER CODEFORMAT 0 SIND NICHT PAARIG ODER DAS CODEFORMAT TRIFFT NICHT ZU.
**** 6 ****	FUER EINE WORTKETTE WIRD EINE BEGRENZUNG ANGEGEBEN, SO DASS DIE OBERE GRENZE KLEINER IST ALS DIE UNTERE.
**** 7 ****	IN DEN POSITIONSANGABEN UNTER CODEFORMAT 1 WIRD KEIN ZAHLENTRIPEL ANGEGEBEN ODER DAS CODEFORMAT TRIFFT NICHT ZU.
**** 8 ****	FUER EINE ZEICHENKETTE WIRD EINE BEGRENZUNG ANGEGEBEN (DIE LETZTEN BEIDEN ZAHLEN EINES TRIPELS), SO DASS DIE OBERE GRENZE KLEINER IST ALS DIE UNTERE.
**** 9 ****	ES WURDE MEHR ALS EIN OBLIGATORISCHES MERKMAL CODIERT.

```
MERKMALSKONTROLLISTE ZUR PARALLELCODIERUNG                TEXT: xaa                SEITE    1
================================================================================================

SATZ-NUMMER:     1
-----------------
xxxxxxaa das Hauptfanggebiet für Schwertfische im z+ Mittelmeer46 +z liegt eigentlich in der z+ Straße von Messina09 +z .


CODIERUNGSEBENE  1
  NR. MERKMAL                                   TEXTSEQUENZ -

    1 +SPRW+                                    xxxxxxaa
    2 +ART +                                    das
    3 +SUB +                                    Hauptfanggebiet
    4 +PREP+                                    für
    5 +SUB +                                    Schwertfische
    6 +PREP+                                    im
    7 +Z   +                                    z+
    8 +SUB +                                    Mittelmeer46
    9 +Z   +                                    +z
   10 +VRB +STV +                               liegt
   11 +ADJ1+                                    eigentlich
   12 +PREP+                                    in
   13 +ART +                                    der
   14 +Z   +                                    z+
   15 +SUB +                                    Straße
   16 +PREP+                                    von
   17 +SUB +                                    Messina09
**** 1 ****   KARTE: XAA:    1: 1:0:XAA00001020NG+SI+NOM          :2,3          :          **********

**** 2 ****   KARTE: XAA:    1: 1:0:NG+PL+PP+ATT                  ,:4.5         :          **********

**** 4 ****   KARTE: XAA:    1: 1:0:NG+SI+PP+ATT+LOK              :6,9          :          **********
```

Abbildung 12

SATZ-NUMMER: 54

ich war aber heilfroh26 ,+ als wir nun ersten an Bord haben49 +, .

CODIERUNGSEBENE 1

NR.	MERKMAL	TEXTSEQUENZ
883	+PERS+	ich
884	+VRB +STVS+	war
885	+ADV +	aber
886	+ADJ1+	heilfroh26
887	+KONS+TEMP+	als
888	+PERS+	wir
889	+ADV +	nun
890	+ADJ1+	ersten
891	+PREP+	an
892	+SUB +	Bord
893	+VRB +SWV +	haben49

CODIERUNGSEBENE 2

NR.	MERKMAL	TEXTSEQUENZ
894	+NG +1P +SI +NOM +	ich
895	+VG +1P +SI +ID +IPF +AK +	war
896	+NG +1P +PL +NOM +	wir
897	+TEMP+	nun
898	+NG +PP +LOK +	an Bord
899	+VG +1P +PL +ID +PRE +AK +	haben49

Abbildung 13

Im nächsten Schritt kann dann dieses Protokoll manuell bearbeitet werden. Für jede Codierung wird überprüft, ob die ausgewählte Textsequenz und auch die Merkmale selbst zutreffen. Zweckmäßigerweise werden dazu zwei Listen geführt. Die eine Liste enthält die Nummern der Codierungen, die inhaltlich falsch sind und aus dem Datenbestand wieder entnommen werden sollen. In die zweite Liste, ein Ablochschema für Codierungen, werden die zu entnehmenden Codierungen in verbesserter Form, sowie die als formal falsch markierten Codierungen neu aufgenommen. Schließlich kann diese Liste noch um weitere, neu aufzunehmende Codierungen ergänzt werden. (Einzelheiten hierzu siehe Anhang I.)

Nach Abschluß der Korrekturarbeit wird die Nummernliste der zu entnehmenden Codierungen dem Programm PC-40 LOESCHEN UND INVENTARISIEREN VON CODIERUNGEN zugeleitet. Dieses Programm verwendet die bereits aufgenommenen Codierungen, führt darin die gewünschten Streichungen durch und mischt für den Teilnehmer die verbleibenden und demnach richtigen Codierungen in eine Stammdatei zum Text. Auch wenn keine inhaltlichen Fehler in den Codierungen enthalten waren, muß dieses Programm gestartet werden, denn erst dadurch werden die in PC-30 umgewandelten Codierungen endgültig in den Codierungsvorrat für einen Text aufgenommen und stehen dann dort dem Teilnehmer zur Verfügung.

Nun kann sich wieder eine Umwandlung von Codierungen anschließen (PC-30), die sowohl die verbesserten als auch neue Codierungen bearbeiten kann. Schließlich wird auch das dazu gehörige Protokoll korrigierend bearbeitet, und mit der endgültigen Aufnahme richtiger Codierungen in die Stammdatei schließt sich der Kreislauf.

Der Vollständigkeit halber sei schon hier gesagt, daß auch in eine Stammdatei korrigierend

eingegriffen werden kann.

Diese Reihenfolge der Bearbeitung muß nur für die Arbeit zum gleichen Text beibehalten werden; die gleichzeitige Bearbeitung verschiedener Texte nebeneinander wird davon nicht betroffen.

3.6 Manipulation von Textsequenzen durch Positionsangaben

Es besteht keine Notwendigkeit dafür, daß eine durch Positionsangaben spezifizierte Textsequenz auch in dieser Form im Satz selbst erscheint. Tatsächlich läßt sich mit Hilfe der Positionsangaben jede mögliche Zeichenkette bilden, wenn nur die nötigen Konstituenten im Satz enthalten sind. Die endlich resultierende Textsequenz nämlich wird über die Positionsangaben aus der Aufreihung der intern begrenzten Zeichenketten erzeugt.

Betrachten wir als Beispiel den Satz

```
ENDLICH WAREN SIE ANGEKOMMEN
O1      O2    O3  O4
```

Wir konstruieren daraus über Codeformat O

```
4,4,2,2              ANGEKOMMEN WAREN
3,3,2,2,4,4          SIE WAREN ANGEKOMMEN
```

oder über Codeformat 1

```
1,1,3,3,3,3          ENDE
```

4,1,2,4,5,10	ANKOMMEN
4,5,11,3,1,4,1,1,7	KOMMEN SIE ENDLICH
2,1,4,1,8,11,3,4,14	WARE WAR ANGEKOMMEN

(Das zweite Tripel: 1,8,11 liefert das Leerzeichen vor WAR, deshalb die Zahlung vom ersten Wort aus, so auch im dritten Tripel.)

Eine mögliche Anwendung für derartige Manipulationen könnte ein parallelcodiertes Grundformenlexikon sein, in dem als Einträge Grundformen sowie mögliche wortbildenden Elemente enthalten sind. Über Positionsangaben könnten dann aus diesen "Wörtern" oder Zeichenketten Formen gebildet werden, deren Eigenschaften in Codierungsangaben festgehalten sind.

4. DIE INTERNE INFORMATIONSVERSCHLÜSSELUNG DER PARALLELCODIERUNG

4.1 B i n ä r c o d e s

Alle digitalen Rechner arbeiten im sogenannten binären Modus, das heißt auf der Basis physikalischer Größen, die genau zwei Zustände unterscheiden können. Ein Ferritring eines Magnetkernspeichers beispielsweise ist entweder in der einen oder in der anderen Richtung magnetisiert, eine Schaltung läßt Strom fließen oder nicht.

Ein derartiger Baustein zur Darstellung von Informationen heißt "Bit". Die beiden möglichen Zustände, die ein Bit annehmen kann, beschreiben wir mit den Zeichen 0 und 1 und sprechen von Binärzeichen.

Nach BAUER - GOOS[8] ist ein Code "eine Vorschrift zur Abbildung eines Zeichenvorrats in einen anderen Zeichenvorrat (oder Wortvorrat)". Die Begriffe Zeichen und Zeichenvorrat als Menge definierter Zeichen setzen wir an dieser Stelle als bekannt voraus. Da wir uns einer digitalen Rechenanlage bedienen, interessieren uns besonders Codes, für die die Bildmenge aus Wörtern über dem Zeichenvorrat {0,1} besteht. Jeder solche Code, der sich also nur zweier Zeichen bedient, heißt Binärcode.

Eine einfachste Vorschrift in diesem Sinne ist ein Code, der einer wahren Aussage das Zeichen 1, einer falschen Aussage das Zeichen 0 zuordnet. Es ist klar, daß dabei die Urbildmenge beliebig viele Elemente enthält (nämlich alle Aussagen), der Bildbereich aber nur zwei Elemente. Ein durch diesen Code erzeugtes Codewort (das Zeichen 0

oder 1) kann demnach nicht die ursprüngliche Aussage liefern, die Abbildung ist nicht umkehrbar.

Ein weiterer bekannter Code ist der Morse-Code, der sich der binären Zeichen "-" und "." bedient und in Codewörtern von unterschiedlicher Zeichenanzahl (Länge) die gebräuchlichsten Schriftzeichen verschlüsselt. Dieser Code ist umkehrbar, aber nur dann, wenn die Trennung zweier Codewörter bekannt ist.

--.-	= Q	
- -.-	= T K	
-- .-	= M A	Gleiche Zeichenfolgen mit verschiedener Trennung im Morse-Code
--. -	= G T	

Implizit wird also ein drittes Zeichen verwendet, ein Trennzeichen, das im Morse-Code als Zeitintervall verstanden wird, in dem keine Zeichen übertragen werden.

Implizit muß fast jeder Binärcode ein Trennzeichen verwenden, falls er mehr als nur eine triviale Unterscheidung, etwa wie im ersten Beispiel nach "wahr" und "falsch", zulassen will. Als ein solches Trennzeichen dient die Länge eines Codewortes, also die Anzahl zusammenhängender Bits, die als Einheit behandelt wird. Im allgemeinen nennt man dann einen Binärcode nach dieser Länge.

Ein 8-Bit-Code ist in diesem Sinne der EBCDI-Code (extended binary coded decimal interchange). Er bildet beispielsweise das Zeichen "A" des natürlichen Alphabets in die 8-Bitkette 11000001 ab. Der Bildbereich dieses Codes sind geordnete 8-Tupel (Codewörter), die aus dem Zeichenvorrat {0,1}

gebildet werden.

Besteht ein Codewort aus n Binärzeichen, dann hat die Menge aller möglichen Codewörter genau 2^n Elemente. Ein solcher Code kann dann umkehrbar angegeben werden, wenn die Anzahl der Urbilder unter diesem Code nicht größer ist als 2^n.

Da in unserem Zusammenhang nur ein umkehrbarer Code interessiert - die zu verschlüsselnde Information soll wieder rekonstruierbar sein -, kann schließlich auch die Definition bei GROSS - LENTIN[9] zugrundegelegt werden, wo die Abbildungsvorschrift als Codierung und die durch die Codierung erzeugten Codewörter als der Code bezeichnet sind.

4.2 32 - Bit - Code der Parallelcodierung

Die Festlegung auf 8 Bits im EBCDI-Code beruht auf dem technischen Aufbau von Rechenahlagen, die man als Byte-Maschinen bezeichnet. Dort ist die kleinste Einheit des Kernspeichers, die adressiert werden kann, ein Byte zu 8 Bits. Im Gegensatz dazu greifen Wort-Maschinen auf ein ganzes Speicherwort zu. In der Regel werden hierbei Zeichen durch einen 6- oder 7-Bit-Code verschlüsselt, wobei ein Speicherwort mehrere Zeichen enthalten kann. Die bestehende Version des Systems Parallelcodierung ist für eine Byte-Maschine konzipiert. Die Einheit, die von einer höheren Programmiersprache adressiert werden kann, ist das Speicherwort zu vier Bytes, also 32 Bits.

Es besteht für die Codierung der eine Textsequenz beschreibenden Merkmale offensichtlich die Möglichkeit, die mnemonischen Kurzwörter selbst

als Code zu verwenden. Ein solches Codewort könnte in einem Speicherwort niedergelegt werden, da es höchstens vier Zeichen enthält. Da die Anzahl der Merkmale, die eine Codierungsangabe bilden, aber variabel ist, müßten mindestens so viele Speicherwörter reserviert werden, wie Merkmale auftreten können. Führt man hier eine Beschränkung auf maximal 10 mnemonische Kurzwörter pro Codierungsangabe ein, so müßten also 40 Bytes bereitgestellt werden. Diese Anzahl beeinflußt aber insbesondere die Geschwindigkeit der Ein/Ausgabe der Daten vom Kernspeicher auf ein externes Trägermedium (beispielsweise ein Magnetband) und umgekehrt. Bedenkt man, daß während einer einzigen Ein/Ausgabe-Operation Tausende von Rechenoperationen durchgeführt werden können, erscheint es sinnvoll, eine rechenintensive Codierung zu wählen, welche die Zeit für die Ein/Ausgabe gering hält. Eine solche Codierung legt ihren Code zweckmäßigerweise auf ein Speicherwort, also auf 32 Bits aus.

Die zu verschlüsselnden Informationen sind dabei die möglichen Codierungsangaben zu einer Textsequenz auf einer unter dem Analysemodell definierten Ebene. Die einzelnen Konstituenten dieser Codierungsangaben sind klassifiziert und als mnemonische Kurzwörter im Codeumsetzer niedergelegt.

Eine Möglichkeit der Codierung wäre die Bildung alle Kombinationen klassenfremder mnemonischer Kurzwörter und ihre Abbildung in die Menge der 32-stelligen Permutationen über {0,1}. Neben dem großen Speicheraufwand hat das Verfahren den Nachteil, daß Paarungen von Kurzwörtern, die in Wirklichkeit niemals auftreten, gebildet werden wie beispielsweise

VERB+NOMINATIV

In der Parallelcodierung verwenden wir ein ähn-

liches Verfahren, ohne daß aber explizit alle Kombinationen gebildet werden, so daß auch keine wirklichkeitsfremden Elemente eliminiert werden müssen. Als Bildvorrat wählen wir die Menge der 32-stelligen Permutationen über {0,1}, das sind genau $2^{32} = 4 \cdot 294 \cdot 967 \cdot 296$ Bildelemente. Eine solche Anordnung von 0 und 1 auf 32 Stellen verstehen wir als Bitmuster (Binärmuster, 32-Bitkette) eines Speicherwortes.

Es muß nun eine Vorschrift angegeben werden, wie ein solches Bitmuster aus den Codierungsangaben erzeugt wird. Dazu betrachten wir eine Klasse von Merkmalen innerhalb einer Ebene, die k Elemente umfassen möge. Dann gibt es eine Zahl n derart, daß gilt

$$2^{n-1} < k \leq 2^n$$

und wir wissen, daß mindestens n Bits zur Verschlüsselung der k Merkmale nötig sind. Ein solches n läßt sich leicht über sukzessives Ausrechnen der Potenzen von 2 und Abgrenzen gegenüber k bestimmen. Es sei $n = n_1$ für eine erste Klasse von Merkmalen bestimmt. Wir reservieren nun die ersten n_1 Bits des zu belegenden Speicherwortes für die Codierung der Merkmale der ersten Klasse.

Zu einer zweiten Klasse läßt sich ebenfalls zu der vorliegenden Anzahl ihrer Elemente ein n mit gleichen Eigenschaften berechnen wie oben, und zur Verschlüsselung der Merkmale dieser Klasse verwenden wir die nächsten n_2 Bits des Speicherwortes. Es ist klar, wie nun auch die restlichen Merkmale der Ebene klassenweise verschlüsselt werden. Schließlich vereinbaren wir noch, daß die möglicherweise ungenutzt verbleibenden Bits mit 0 aufgefüllt werden.

Eine Klasse von Merkmalen möge beispielsweise 6 Elemente umfassen, und wir finden n = 3, denn

$$2^2 = 4 < 6 \leq 8 = 2^3$$

Die einzelnen Merkmale werden nun numeriert, beginnend mit 1. Diese Indizes lassen sich in binärer Schreibweise darstellen, und zwar wollen wir dazu jede solche Binärzahl durch führende Nullen auf genau n Stellen ergänzen. Für obiges Beispiel erhalten wir:

1. Merkmal	001
2. Merkmal	010
3. Merkmal	011
4. Merkmal	100
5. Merkmal	101
6. Merkmal	110

Die derart erzeugten Binärzahlen lassen sich so als Bitmuster für die Codierung der einzelnen Merkmale einer Klasse verwenden. Was hierbei mit führenden Nullen der Beschreibung wegen manipuliert wurde, wird von einer Rechenmaschine automatisch durchgeführt.

So wurde schließlich jedem Merkmal einer Klasse als Index ein W e r t (Binärzahl) zugeordnet. Für jede Klasse ist bekannt, wieviele Bits (L ä n g e) reserviert werden müssen. Die einzelnen Klassen erhalten, resultierend aus ihrer Aneinanderreihung, eine Größe zugeordnet, die wir mit I n z i d e n z bezeichnen, und die angibt, das wievielte Bit der 32-Bitkette des Speicherwortes mit dem letzten Bit einer n-Bitkette, welche die Merkmale einer Klasse verschlüsselt, zusammenfällt.

Wenn die Numerierung der Merkmale zur Bestimmung

der Binärwerte mit 1 beginnt, lassen sich auf n Stellen nur 2^n-1 Merkmale verschlüsseln, weil die n-Bitkette, die nur Nullen enthält, nicht angenommen wird. Da, wie schon erwähnt, führende Nullen, aber auch Teilketten, die nicht durch Merkmale ausgewiesen werden, als Nullen von der Rechenmaschine automatisch gesetzt werden, darf eine Kette, die nur O enthält, nicht mit einem Merkmal belegt werden, weil sonst die Umkehrung der Codierung, also die Rückfindung von Merkmalen aus der verschlüsselnden 32-Bitkette, nicht gewährleistet ist. Nur für die Menge der obligatorischen Merkmale einer Ebene wird die O-Kette verwendet: dort bezeichnet sie das neutrale Element, das angibt, daß kein obligatorisches Merkmal zutrifft, und das automatisch durch das mnemonische Kurzwort "...." ausgewiesen wird.

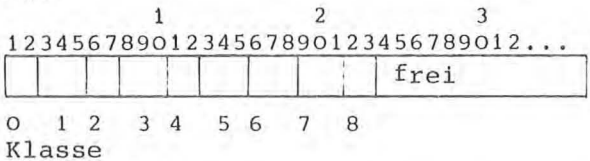
Es folgt daraus, daß zur Bestimmung der Anzahl der benötigten Bitstellen zur Verschlüsselung von Merkmalen einer Klasse die gegebene Anzahl der Merkmale um 1 vermehrt werden muß.

Die folgende Darstellung zeigt für einen möglichen Beispielfall (Ebene 2 der Freiburger Codierungen) die Bestimmung der Teilketten aus der Anzahl der gegebenen Merkmale und ihre Aneinanderreihung in der verschlüsselnden 32-Bitkette:

Klassenindex	gegebene Merkmale		Anzahl benötigter Bits	resultierende Inzidenz
	k	k+1	n	i
O	2	3	2	2
1	4	5	3	5
2	2	3	2	7

Klassen-Index	gegebene Merkmale k	k+1	Anzahl benötigter Bits n	resultierende Inzidenz i
3	6	7	3	10
4	3	4	2	12
5	4	5	3	15
6	6	7	3	18
7	7	8	3	21
8	1	2	1	22

Damit ergibt sich folgender Aufbau des Speicherwortes:



Die Codierung der Merkmale im einzelnen, also die zugeordneten Größen Wert - Länge - Inzidenz werden durch jedes einen Codeumsetzer aufbereitende Programm (PC-10/11/12) protokolliert. Die folgenden Abbildungen 14 bis 18 zeigen diese Codierung für die drei Ebenen der Freiburger Forschungsstelle.

Abbildung 14

CODIERUNG DER 1.EBENE (46 MERKMALE)

KLASSE 0: (34 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	....	0	6	6
	VRB	1	6	6
	VRB1	2	6	6
	VRB2	3	6	6
	VRB3	4	6	6
	VZS	5	6	6
	PTZ1	6	6	6
	PTZ2	7	6	6
	INF	8	6	6
	INFZ	9	6	6
	SUB	10	6	6
	ART	11	6	6
	PERS	12	6	6
	REFL	13	6	6
	POSS	14	6	6
	FRPR	15	6	6
	PRON	16	6	6
	ADJ1	17	6	6
	ADJ2	18	6	6
	ADV	19	6	6
	ADVN	20	6	6
	KONK	21	6	6
	KONS	22	6	6
	PREP	23	6	6
	ANTP	24	6	6
	ZERO	25	6	6
	SPRW	26	6	6
	FR	27	6	6
	E	28	6	6
	P	29	6	6
	Z	30	6	6
	F	31	6	6
	B	32	6	6
	K	33	6	6

KLASSE 1: (2 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	LE	1	2	8
	GRF	2	2	8

Abbildung 15

CODIERUNG DER 1.EBENE (FORTSETZUNG)

KLASSE 2: (4 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	STV	1	3	11
	SWV	2	3	11
	STVS	3	3	11
	FV	4	3	11
KLASSE 3: (2 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	KOMP	1	2	13
	SUP	2	2	13
KLASSE 4: (4 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	MOD	1	3	16
	TEMP	2	3	16
	KAUS	3	3	16
	FIN	4	3	16

Abbildung 16

CODIERUNG DER 2.EBENE (36 MERKMALE)

KLASSE 0: (3 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	....	0	2	2
	VG	1	2	2
	NG	2	2	2
KLASSE 1: (4 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	1P	1	3	5
	2P	2	3	5
	3P	3	3	5
	ANR	4	3	5
KLASSE 2: (2 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	SI	1	2	7
	PL	2	2	7
KLASSE 3: (6 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	PRE	1	3	10
	IPF	2	3'	10
	PER	3	3	10
	PQU	4	3	10
	F1	5	3	10
	F2	6	3	10
KLASSE 4: (3 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	AK	1	2	12
	WP	2	2	12
	SP	3	2	12
KLASSE 5: (4 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	ID	1	3	15
	KJ	2	3	15
	KJW	3	3	15
	IMP	4	3	15

Abbildung 17

CODIERUNG DER 2.EBENE (FORTSETZUNG)

KLASSE 6: (6 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	NOM	1	3	18
	GEN	2	3	18
	DAT	3	3	18
	AKK	4	3	18
	PP	5	3	18
	IK	6	3	18

KLASSE 7: (7 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	MOD	1	3	21
	TEMP	2	3	21
	KALZ	3	3	21
	TEKA	4	3	21
	KAUS	5	3	21
	LOK	6	3	21
	LEER	7	3	21

KLASSE 8: (1 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	ATT	1	1	22

Abbildung 18

CODIERUNG DER 3.EBENE (17 MERKMALE)

KLASSE 0: (10 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	....	0	4	4
	HS	1	4	4
	EHS	2	4	4
	AH	3	4	4
	NS	4	4	4
	IS	5	4	4
	DV	6	4	4
	SS	7	4	4
	KS	8	4	4
	PT	9	4	4

KLASSE 1: (2 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	EST	1	2	6
	NEST	2	2	6

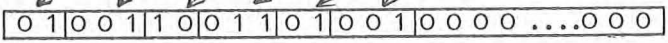
KLASSE 2: (5 MERKMALE)	MERKMAL	WERT	LAENGE	INZIDENZ
	SELE	1	3	9
	AUES	2	3	9
	AVS	3	3	9
	ANS	4	3	9
	AVK	5	3	9

Eine Codierungsangabe, etwa auf der 2. Ebene, gegeben als

VG+1P+PL+PER+ID+AK

wird damit durch diese Codierung umgewandelt in die 32-Bitkette:

Kurzwort	VG	1P	PL	PER	ID	AK
Wert	1	1	2	3	1	1
Länge	2	3	2	3	3	2
Inzidenz	2	5	7	10	15	12
Bitkette	01	001	10	011	001	01



Bit 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
1 3

Es ist dabei gleichgültig, in welcher Reihenfolge die einzelnen mnemonischen Kurzwörter aufgeführt werden. Durch die ihnen zugeordnete Inzidenz werden sie an die vorgesehene Stelle innerhalb der 32-Bitkette eingebettet.

Bis jetzt wurde gezeigt, auf welche Weise die Codierung als Abbildung der Merkmale in die Menge der möglichen 32-Bitketten funktioniert. Da wir aber auch die Umkehrbarkeit dieser Abbildung fordern, also die Möglichkeit zur Decodierung erwarten, muß die Abbildungsvorschrift entsprechend erweitert werden.

Abbildung 19

CODEUMSETZER (INSGESAMT 99 MERKMALE)

	EBENE 1	EBENE 2	EBENE 3	EBENE 4	EBENE 5	EBENE 6	EBENE 7	EBENE 8	EBENE 9	EBENE 10
POINTER ZU DEN EBENEN:	1	47	83	0	0	0	0	0	0	0
POINTER ZUR 1.INZIDENZ:										
BIT 1	0	0	0	0	0	0	0	0	0	0
BIT 2	0	68	0	0	0	0	0	0	0	0
BIT 3	0	0	0	0	0	0	0	0	0	0
BIT 4	0	0	83	0	0	0	0	0	0	0
BIT 5	0	55	0	0	0	0	0	0	0	0
BIT 6	2	0	92	0	0	0	0	0	0	0
BIT 7	0	49	0	0	0	0	0	0	0	0
BIT 8	15	0	0	0	0	0	0	0	0	0
BIT 9	0	0	84	0	0	0	0	0	0	0
BIT 10	0	53	0	0	0	0	0	0	0	0
BIT 11	19	0	0	0	0	0	0	0	0	0
BIT 12	0	54	0	0	0	0	0	0	0	0
BIT 13	23	0	0	0	0	0	0	0	0	0
BIT 14	0	0	0	0	0	0	0	0	0	0
BIT 15	0	48	0	0	0	0	0	0	0	0
BIT 16	1	0	0	0	0	0	0	0	0	0
BIT 17	0	0	0	0	0	0	0	0	0	0
BIT 18	0	52	0	0	0	0	0	0	0	0
BIT 19	0	0	0	0	0	0	0	0	0	0
BIT 20	0	0	0	0	0	0	0	0	0	0
BIT 21	0	47	0	0	0	0	0	0	0	0
BIT 22	0	50	0	0	0	0	0	0	0	0
BIT 23	0	0	0	0	0	0	0	0	0	0
BIT 24	0	0	0	0	0	0	0	0	0	0
BIT 25	0	0	0	0	0	0	0	0	0	0
BIT 26	0	0	0	0	0	0	0	0	0	0
BIT 27	0	0	0	0	0	0	0	0	0	0
BIT 28	0	0	0	0	0	0	0	0	0	0
BIT 29	0	0	0	0	0	0	0	0	0	0
BIT 30	0	0	0	0	0	0	0	0	0	0
BIT 31	0	0	0	0	0	0	0	0	0	0
BIT 32	0	0	0	0	0	0	0	0	0	0

Abbildung 20

INHALT DES CODEUMSETZERS

LFD. INDEX	MERKMAL	WERT	LAENGE	INZIDENZ	POINTER	BAUM <	BAUM >
1	MOD	1	3	16	14	2	3
2	FR	27	6	6	3	4	5
3	SUB	10	6	6	4	25	26
4	ANTP	24	6	6	5	6	7
5	K	33	6	6	6	15	16
6	ADJ2	18	6	6	7	8	9
7	E	28	6	6	8	11	12
8	ADJ1	17	6	6	9	0	0
9	ADV	19	6	6	10	0	10
10	ADVN	20	6	6	11	0	0
11	ART	11	6	6	12	0	13
12	F	31	6	6	13	0	14
13	B	32	6	6	16	0	0
14	FIN	4	3	16	21	0	0
15	GRF	2	2	8	24	17	18
16	KONK	21	6	6	17	21	22
17	FRPR	15	6	6	18	0	19
18	INF	8	6	6	20	0	20
19	FV	4	3	11	34	0	0
20	INFZ	9	6	6	22	0	0
21	KAUS	3	3	16	37	0	23
22	KONS	22	6	6	25	0	24
23	KOMP	1	2	13	39	0	0
24	LE	1	2	8	24	0	0
25	PTZ1	6	6	6	26	27	28
26	VRB2	3	6	6	27	37	38
27	POSS	14	6	6	28	29	30
28	SPRA	26	6	6	29	33	34
29	P	29	6	6	30	0	31
30	PREP	23	6	6	31	0	32
31	PERS	12	6	6	32	0	0
32	PRON	16	6	6	33	0	0
33	PTZ2	7	6	6	35	0	35
34	STV	1	3	11	36	0	36
35	REFL	13	6	6	38	0	0
36	STVS	3	3	11	41	0	0
37	TEMP	2	3	16	37	39	40
38	Z	30	6	6	40	43	44
39	SUP	2	2	13	39	0	41
40	VRB	1	6	6	42	0	42
41	SWV	2	3	11	41	0	0
42	VRB1	2	6	6	43	0	0
43	VRB3	4	6	6	44	0	45
44	ZERO	25	6	6	45	0	46
45	VZS	5	6	6	46	0	0
46	....	0	6	6	46	0	0
47	LOK	6	3	21	51	48	49
48	ID	1	3	15	59	50	51
49	S1	1	2	7	72	66	67
50	ATT	1	1	22	50	52	53

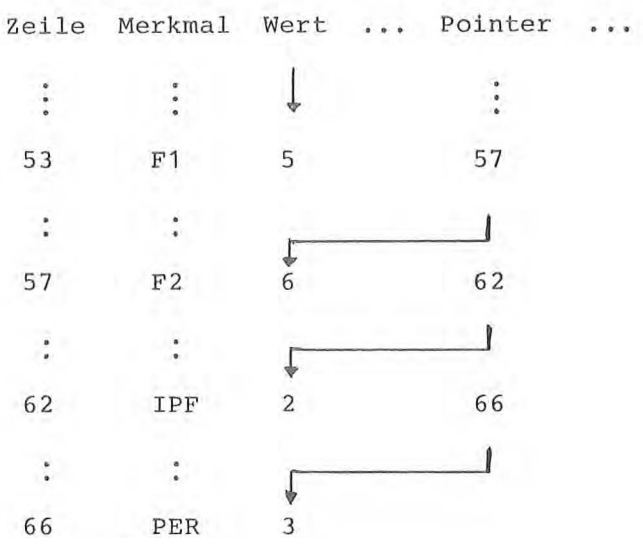
Abbildung 21

INHALT DES CODEUMSETZERS

LFD.	INDEX	MERKMAL	WERT	LAENGE	INZIDENZ	POINTER	BAUM <	BAUM >
	51	KALZ	3	3	21	63	59	60
	52	AKK	4	3	18	56	54	55
	53	F1	5	3	10	57	56	57
	54	AK	1	2	12	67	46	46
	55	ANR	4	3	5	76	46	46
	56	DAT	3	3	18	58	46	46
	57	F2	6	3	10	62	46	58
	58	GEN	2	3	18	61	46	46
	59	IMP	4	3	15	60	61	62
	60	KJ	2	3	15	64	63	64
	61	IK	6	3	18	69	46	46
	62	IPF	2	3	10	66	46	46
	63	KAUS	5	3	21	65	46	46
	64	KJW	3	3	15	64	46	65
	65	LEER	7	3	21	70	46	46
	66	PER	3	3	10	73	68	69
	67	WP	2	2	12	77	75	76
	68	NG	2	2	2	79	70	71
	69	PP	5	3	18	71	72	73
	70	MOD	1	3	21	75	46	46
	71	WOM	1	3	18	71	46	46
	72	PL	2	2	7	72	46	46
	73	PQU	4	3	10	74	46	74
	74	PRE	1	3	10	74	46	46
	75	TEKA	4	3	21	78	77	78
	76	2P	2	3	5	80	80	81
	77	SP	3	2	12	77	46	46
	78	TEMP	2	3	21	78	46	79
	79	VG	1	2	2	82	46	46
	80	1P	1	3	5	81	46	46
	81	3P	3	3	5	81	46	82
	82	....	0	2	2	82	46	46
	83	HS	1	4	4	85	84	85
	84	AVK	5	3	9	86	86	87
	85	NS	4	4	4	87	93	94
	86	ANS	4	3	9	89	88	89
	87	DV	6	4	4	88	90	91
	88	AH	3	4	4	91	82	82
	89	AUES	2	3	9	90	82	82
	90	AVS	3	3	9	94	82	82
	91	EHS	2	4	4	93	82	92
	92	EST	1	2	6	96	82	82
	93	KS	8	4	4	95	95	96
	94	SELE	1	3	9	94	97	98
	95	IS	5	4	4	97	82	82
	96	WEST	2	2	6	96	82	82
	97	PT	9	4	4	98	82	82
	98	SS	7	4	4	99	82	99
	99	....	0	4	4	99	82	82

Neben den Codierungsdaten, nämlich Wert, Länge und Inzidenz, werden den Merkmalen, die im Codeumsetzer eingebracht sind, noch Strukturdaten zugeordnet, die das Codieren (über eine Baumstruktur) und das Decodieren (über eine Liste) beschleunigen helfen. Zu diesen Strukturdaten gehören Pointer, die in die Nähe der jeweils relevanten Informationen verweisen. Mit Hilfe dieser Pointer wird für jede Ebene der Teilkettenaufbau der verschlüsselnden 32-Bitketten sichtbar gemacht (siehe Abbildung 19, Spalten unter "Pointer zur 1. Inzidenz"). Daraus lassen sich Länge und Inzidenz der einzelnen Teilketten bestimmen. Aus einer verschlüsselnden 32-Bitkette kann dann jede sinnvolle Teilkette entnommen werden, das heißt: es kann der in der Teilkette niedergelegte Wert bestimmt werden. Der im Codeumsetzer enthaltene Pointer verweist nun auf das erste Merkmal, das zu der jeweiligen Klasse gehört. Stimmt der dort niedergelegte Wert nicht mit dem ermittelten überein, wird über den Listenverweis gegen das nächste Element der Klasse verglichen, bis schließlich das zutreffende mnemonische Kurzwort gefunden ist.

Wird beispielsweise die auf Seite 74 codierte Bitkette decodiert, etwa für die dritte Klasse, erhalten wir (siehe auch Abbildungen 19 bis 21) auf Ebene 2 den Bitabstand 10-7 = 3 als Länge mit der Inzidenz 10. Es kann nun die Teilkette entnommen werden, sie liefert den Wert 3. Der Pointer verweist in die 53-te Zeile des Codeumsetzers, in der der Wert 5 ausgewiesen ist. Also wird über den Listenverweis (Spalte "Pointer") weitergesucht:


Zeile
Merkmal
Wert
...
Pointer
...
53
F1
5
57
57
F2
6
62
62
IPF
2
66
66
PER
3

An dieser Stelle stimmt der Wert mit dem vorliegenden überein, also ist das zugehörige mnemonische Kurzwort das Merkmal "PER".

In dieser Weise ist also gewährleistet, daß die Decodierung einer verschlüsselnden 32-Bitkette alle die mnemonischen Kurzwörter liefert, die in die Codierung eingegangen sind. Damit kann in der Parallelcodierung stets mit einer lesbaren Form der jeweiligen Information gearbeitet werden, wobei dennoch eine maschinengerechte interne Darstellung vorliegt.

4.3 Schwierigkeiten bei der Codierung

Die Festlegung auf 32 Bits in der Parallelcodierung setzt eine willkürliche, aus der verwendeten Rechenanlage resultierende Beschränkung. Es besteht nicht die Möglichkeit, diese Grenze zu überschreiten, es sei denn, es wird verändernd in die Programme eingegriffen.

Solange nun eine Inzidenz keinen Wert größer als 32 annimmt, bleibt das Verfahren problemlos. Erst wenn auf einer Ebene so viele Klassen definiert sind, daß die lineare Aufreihung der verschlüsselnden Teilketten über die 32-Bitkette hinaus reicht, muß ein Weg gesucht werden, der das Verfahren dennoch praktikabel macht.

Eine erste Möglichkeit ist die Verteilung der Vielzahl von Klassen auf mehrere Ebenen. Dieser Weg ist der einfachste; allerdings muß hier schon berücksichtigt werden, daß ein Retrieval auf maximal 3 Ebenen zugleich zugreift. Es kann demnach sein, daß durch diese Verteilung der Informationszugriff zusätzlich noch eingeschränkt

wird. Auch die manuelle Erstellung einer Codierung wird sich dabei erschweren: was eigentlich auf einer Lochkarte allein stehen sollte, muß jetzt nämlich auch verteilt codiert werden.

Im allgemeinen wird es so sein, daß nur bestimmte Klassen fakultativer Merkmale innerhalb einer Ebene sich auf ein obligatorisches Merkmal $x \in \overline{M}$ dieser Ebene beziehen und nicht alle zugleich. Das bedeutet: In einer verschlüsselnden 32-Bitkette sind nicht immer alle Teilketten zugleich interessant, sondern ein Teil von ihnen besteht nur aus Null-Ketten. Seien x_1, x_2, ... obligatorische Merkmale und a, b, ... Repräsentanten von Klassen fakultativer Merkmale auf einer Ebene, und seien beispielsweise folgende Beiordnungen an fakultativen Merkmalen möglich:

x_1 :: a, b, d, e

x_2 :: a, b, c

x_3 :: a, c, e, f

x_4 :: a, b, c, e

dann läßt sich die lineare Aufreihung der verschlüsselnden Teilketten in folgendem Schema beschreiben, wobei "x" die möglichen Beiordnungen markiere:

	a	b	c	d	e	f
x_1	x	x		x	x	
x_2	x	x	x			
x_3	x		x		x	x
x_4	x	x	x			

Man wird bestrebt sein, eine vollständige Ausnutzung aller Bits zu erreichen, ohne daß Stellen ohne Bedeutung verbleiben. Eine solche Verdichtung läßt sich mit diesem Schema einfach organisieren. Ohne Beschränkung der Allgemeinheit sei jeder Beiordnung a, b, ... , also jeder Klasse fakultativer Merkmale die gleiche Länge zugeordnet.

	a	b	c	d	e	f
x_1	x	x		x	(x)	
x_2	x	x	x			
x_3	x		x		(x)	(x)
x_4	x	x	x		(x)	

Man verfahre folgendermaßen:

1. Man bestimme die Beiordnungen, die sich auf nur ein obligatorisches Merkmal beziehen und fülle damit Lücken in der gleichen Zeile (ausgezogene Pfeile).

2. Man verfahre ebenso für Paare, Tripel usw. von Beiordnungen und fülle damit Lückenpaare, usw. (gestrichelte Pfeile).

Damit erreichen wir eine Verdichtung des Schemas und also eine Reduktion der Anzahl der benötigten Bits. In dem beschriebenen Beispiel konnte dadurch das Schema um zwei Spalten (e und f) verringert werden, nur eine Stelle bleibt naturgemäß unbesetzt: (x ,d).

Falls den Beiordnungen a, b, ... verschiedene Anzahlen von Bits entsprechen, läßt sich dies bei dieser Verdichtung einfach dadurch berücksichtigen, daß jeder Beiordnung eine bestimmte Spaltenbreite in Abhängigkeit der für diese Beiordnung nötigen Bits zugeordnet wird. Der Verdichtungsvorgang muß dann beim Umordnen diese Breite berücksichtigen.

Damit aber wird die Interpretation einer fakultativen Beiordnung abhängig vom übergeordneten obligatorischen Merkmal. Der Codeumsetzer der Parallelcodierung enthält jedoch keine Daten, welche eine solche Interpretation zulassen.

Dennoch können wir von diesem Verfahren Gebrauch machen, wenigstens von der zugrundeliegenden Idee, nämlich dann, wenn die Verdichtung so gehandhabt werden kann, daß der Inhalt einer Spalte des Schemas vollständig in genau einer anderen Spalte untergebracht werden kann und keine Aufteilung nach mehreren Spalten vorgenom-

men wird.

Bezogen auf die Klassifizierung des Merkmalsvorrats bei der Erstellung des Codeumsetzers heißt das, daß zwei oder mehr miteinander unvereinbare Klassen von Merkmalen zu einer zusammengefaßt werden. Der so resultierende Klassenumfang benötigt in der Regel weniger Bits zur Codierung, als für die Codierung der Klassen im einzelnen verwendet werden müßten.

Seien K_1 und K_2 zwei Klassen von Merkmalen, die nicht zugleich zutreffen können, und ihr Umfang sei k_1 und k_2. Ohne Einschränkung der Allgemeinheit sei $k_1 \geq k_2$. Zu k_1 gibt es dann eine Zahl n_1 derart, daß

$$2^{n_1-1} < k_1 \leq 2^{n_1} \qquad (*)$$

und analog ein n_2 für k_2. Werden K_1 und K_2 zu einer Klasse K zusammengefaßt, die dann k_1+k_2 Elemente enthält, sind zur Codierung dieser Klasse nur n_1+1 Bits nötig.

Nach (*) gilt nämlich:

$$\begin{aligned} \mathrm{ld}\ k_1 &\leq n_1 \\ \mathrm{ld}\ k_1 + 1 &\leq n_1 + 1 \\ \mathrm{ld}\ k_1 + \mathrm{ld}\ 2 &\leq n_1 + 1 \\ \mathrm{ld}\ (2 \cdot k_1) &\leq n_1 + 1 \end{aligned}$$

ld $\triangleq$ logarithmus dualis

und es ist $k_1 + k_2 \leq 2 \cdot k_1$, so nämlich war k_1 gewählt.

Benötigen also K_1 und K_2 in getrennter Verschlüsselung zusammen $n_1 + n_2$ Bits, so sind, falls beide Klassen zusammengefaßt werden können, schließlich höchstens $\max\{n_1,n_2\}+1$ Bits nötig.

Ein Beispiel illustriert das Verfahren:

$$k_1 = 5 \Rightarrow n_1 = 3$$

$$k_2 = 2 \Rightarrow n_2 = 2$$

$$k_1 + k_2 = 7 \Rightarrow n = 3 < 3 + 2$$

Betrachten wir die Ebene 2 der Freiburger Codierungen. Wir können annehmen, daß die Merkmale der Klassen 1 (Information zur Person) und 6 (Kasusinformation) sich nicht zugleich auf ein obligatorisches Merkmal beziehen können. Fassen wir diese beiden Klassen zu einer zusammen, resultiert folgende Codierung der Klasse:

Merkmal:	1P	2P	3P	ANR	NOM	GEN	DAT	AKK	PP	IP
Wert :	1	2	3	4	5	6	7	8	9	10
Länge :	4	4	4	4	4	4	4	4	4	4

Insgesamt werden damit nur 4 Bits zur Verschlüsselung benötigt, während die Auftrennung in zwei Klassen 3 + 3 = 6 Bits fordert.

Falls also die konzipierte Klassifizierung eines Merkmalsvorrats auf einer Ebene zu Schwierigkeiten führt, läßt sich durch Zusammenfassung von Klassen, aus denen nicht zugleich Merkmale zutreffen können, im allgemeinen die Codierbarkeit erreichen.

Ein ähnlicher Ausweg bietet sich auch an, falls Klassen zusammengefaßt werden müßten, deren Merkmale aber zugleich zutreffen können. Man bilde eine Klasse, deren Elemente Paare von solchen Merkmalen sind und belege die sich ergebenden Paare mit mnemonischen Kurzwörtern. Haben zwei Klassen k_1 bzw. k_2 Elemente, dann sind $k_1 \cdot k_2$ Paare möglich. Dabei werden nicht alle Paarbildungen sinnvoll sein, wie etwa

KONJUNKTIV + FUTUR II

so daß schließlich auch hierbei ein Merkmalsumfang resultiert, der weniger Bits zur Codierung benötigt, als wenn im einzelnen codiert würde.

Für das obige Zahlenbeispiel ergibt sich in diesem Fall:

$$k_1 \cdot k_2 = 15 \Rightarrow n = 4 < 2 + 3$$

In beiden Fällen wird von der Tatsache Gebrauch gemacht, daß mit wachsendem Merkmalsumfang die Anzahl der benötigten Bits nicht linear mitwächst, sondern nur logarithmisch (zur Basis 2).

5. DIE DATEN DER PARALLELCODIERUNG

5.1 Aufbau der Datenträger

Wie in der Einleitung aufgezeigt, bilden die Codierungen zu einem Text einen eigenständigen Datenpool. Diese Daten enthalten nicht den Text oder die codierten Textsequenzen selbst, sondern ermöglichen über die Positionsangaben den Zugriff auf die bezeichneten Sequenzen. So sind also zwei Datenträger zugleich zu betrachten.

1) Der Text

Die größte Einheit eines Textes, auf die eine Parallelcodierung zugreifen kann, ist der Satz (Record), der aus einer beliebigen Zeichenfolge variabler Länge besteht, aber nicht mehr als 5000 Zeichen enthält. Jedem Record sind zugeordnet ein Längenzähler für den Record selbst und ein Folgezähler, der die Aufeinanderfolge der Records fortlaufend numeriert. Eine beliebige Anzahl von aufeinanderfolgenden Records bildet einen Text (File). Der erste Record enthält nur 3 Zeichen, die den Textschlüssel bezeichnen. Ein solcher Text ist im allgemeinen auf einem Magnetband gespeichert; dort können sich auch mehrere Texte als getrennte Files hintereinander befinden.

2) Die Codierungen

Eine Codierung, wie sie aus den Eingabedaten erzeugt wird, stellt sich dar als ein Record fester Länge mit folgendem Aufbau:

- Satznummer; sie verweist auf den zugehörigen Record des Textes
- Ebenenindex der Codierung
- Codierungsangabe als 32-Bitkette
- Anzahl relevanter Paare von Positionsangaben
- 6 Paare von Positionsangaben, die Anfang bzw. Ende von Textsequenzen des zugehörigen Textsatzes bezeichnen.

Die Aufeinanderfolge solcher Codierungen ist in der gleichen Reihenfolge wie der Text selbst sortiert, nämlich nach Satznummern. Innerhalb gleicher Satznummern wird dann weiter nach den Ebenenindizes, für gleiche Ebenenindizes nach der ersten Positionsangabe sortiert.

Werden nun Text und Codierungen zum Text zugleich bearbeitet, bewirkt das Auftreffen auf einen neuen Satz das Spulen der zweiten Datei auf diesen Satz, so daß immer Zusammengehöriges greifbar ist.

5.2 Datenkompatibilität

Da die Programme, die das System Parallelcodierung bilden, alle in Fortran geschrieben sind, und zwar in einem Sprachumfang, der auf den meisten Anlagen entsprechender Größenordnung erklärt ist, genügt es, an dieser Stelle nur die Kompatibilität in den Daten zu betrachten.

Die im Rahmen der Textcodierung erstellten Da-

ten sind austauschbar überall dort, wo mit Speicherwörtern zu 32 Bits gearbeitet wird. Schwierigkeiten können sich ergeben, falls weniger Bits zur Verfügung stehen, wenn nämlich die vorhandene Anzahl die Codierung nicht erlaubt, weil eine Inzidenz darüberhinausgreifende Werte annimmt. Tritt dieser Fall aber nicht ein, können Daten problemlos ausgetauscht werden, ebenso auch, falls auf einer anderen Rechenanlage mehr als 32 Bits im Speicherwort zur Verfügung stehen.

In beiden Fällen aber, in denen die Bitanzahl nicht übereinstimmt, ist für die Datenübernahme eine Umcodierung erforderlich. Die unformatiert gespeicherten Daten müssen formatiert aufbereitet werden, bevor sie beim Empfänger wieder unformatiert umgesetzt den Verarbeitungsprogrammen zugeführt werden können.

Die anzustrebende Kompatibilität stellt damit im wesentlichen nur Forderungen an die Lesbarkeit formatiert beschriebener Datenträger.

Die folgende Abbildung 22 stellt die Anzahlen der Bits pro Speicherwort für einige Rechenanlagen gegenüber. (Die Übersicht ist selbstverständlich unvollständig und die Auswahl subjektiv.)

5.3 Updating von Codierungen

Wir gehen davon aus, daß der zu codierende Text in einer endgültigen Fassung vorliegt. Sonst nämlich müßte im Fall von Veränderungen am Text, welche in die Numerierung der Records eingreifen oder die Aufeinanderfolge der Zeichen inner-

Abbildung 22

HERSTELLER	MODELL	BITS PRO SPEICHERWORT +)
TC	TR 440	48 (W)
CDC	3100 - 3500 (MASTER)	48 (W)
CDC	CYBER 72 - 76	60 (W)
CII	10070 (IRIS 50/80)	32 (W)
IBM	/360, /370	32 (B)
IBM	7090/94	36 (W)
ICL	1900	24 (W)
DEC	10/40 - 70	36 (W)
SIEMENS	4004/35 - 151	32 (B)
UNIVAC	9400	40 (B)
UNIVAC	1106 -1110	36 (W)

+) B = ZUGRIFF KANN AUF EIN EINZELNES BYTE ERFOLGEN (BYTE-MASCHINE)

W = ZUGRIFF ERFOLGT AUF DAS GANZE SPEICHERWORT (WORT-MASCHINE)

halb eines Records - durch Streichen oder Einfügen - betreffen, zugleich die Codierung auf den neuen Stand gebracht werden, da dann der Bezug zum Quelltext nicht mehr fehlerfrei möglich ist.

Codierungen nun, die zu einem Text erstellt werden, können formale und inhaltliche Fehler enthalten. Eingabedaten mit formalen Fehlern werden durch das die Daten aufnehmende Programm PC-30 mit einer entsprechenden Fehlermeldung abgewiesen. Diese Daten hinterlassen keine Spuren auf dem Trägermedium, auf dem die übrigen, formal richtigen Codierungen in intern verschlüsselter Form gespeichert werden. Ein solches Trägermedium, im allgemeinen ein Magnetband, nennen wir Generierungsfile (GENFIL). Die formal richtigen Codierungen, die es nach einer Datenaufnahme enthält, werden im Output des Programms PC-30 (siehe Abbildungen 12 und 13) numeriert wiedergegeben, wobei die ausgewählte Textsequenz als Klartext und nicht über Positionsangaben mitgelistet wird. Bevor nun zum gleichen Text weitere Codierungen aufgenommen werden können, muß dieser Output bearbeitet werden, das heißt, die erfaßten Codierungen müssen auf inhaltliche Fehler geprüft werden.

Dabei werden die Nummern fehlerhafter Codierungen in einem Ablochschema (siehe dazu Anhang I) gesammelt; formal falsche Codierungen brauchen hier natürlich nicht berücksichtigt zu werden, ihnen ist auch keine fortlaufende Nummer zugeordnet. Das Programm PC-40: LÖSCHEN UND INVENTARISIEREN VON CODIERUNGEN übernimmt nun die so gesammelten Nummern und überträgt den Inhalt von GENFIL in einen Stammfile (STMFIL) für Codierungen zu dem bearbeiteten Text, wobei die durch ihre Nummern als fehlerhaft ausgewiesenen Codierungen übergangen werden.

Dabei werden zu übernehmende Codierungen mit

eventuell bereits vorhandenen so zusammengemischt, daß die schon beschriebene Sortierung resultiert. Ist der STMFIL erstellt, so kann mit diesem Datenbestand und dem dazugehörigen Text gearbeitet werden.

Sind die Codierungen von GENFIL nach STMFIL übertragen, können weitere erstellt werden, insbesondere werden die als formal falsch abgewiesenen in korrigierter Form neu eingespielt. Die folgende Abbildung 23 beschreibt den sich ergebenden Kreislauf am Beispiel der Codierung des Textes TXT.

Ist schließlich eine letzte Übertragung von GENFIL nach STMFIL erfolgreich abgeschlossen (sie muß auch dann stattfinden, wenn keine inhaltlichen Fehler vorliegen), und werden keine weiteren neuen Codierungen mehr erstellt, enthält STMFIL alle Codierungen zum Text.

Es kann der Fall eintreten, daß auch ein STMFIL inhaltlich fehlerhafte Codierungen enthält, die bei einem Korrekturvorgang übersehen wurden. Mit Hilfe des Programms PC-50: DECODIEREN UMGEWANDELTER CODIERUNGEN erzeugt man dann ein Protokoll eines STMFIL, das alle je erfaßten Codierungen numeriert wiedergibt (siehe Abbildung 24). Die Nummern der fehlerhaften Codierungen können dann ebenfalls dem Programm PC-40 zugeführt werden, das in diesem Fall lediglich den Inhalt des alten STMFIL, allerdings ohne die beanstandeten Codierungen auf ein neues Band überträgt, ohne daß sortiert und gemischt wird. Nach dieser Übertragung enthält das neu erstellte Band das Etikett STMFIL zum Text, die alte Datei ist damit gelöscht. Selbstverständlich kann ein STMFIL immer wieder erweitert werden und zwar für Codierungen auf all den Ebenen, die in einem Codeumsetzer manifestiert sind.

ERFASSEN, LOESCHEN UND INVENTARISIEREN VON CODIERUNGEN.

(ARBEITSABLAUF)

Abbildung 23

PARALLELCODIERUNG: ABDRUCK DES STAMMBANDES ZU zaa SEITE: 1

SATZ-NUMMER: 1

zxxxxxaa das Hauptfanggebiet für Schwertfische im z+ Mittelmeer46 +z liegt eigentlich in der z+ Straße von Messina09 +z .

LFD.NR	MERKMALE	SQN	TEXTSEQUENZ
1	+SPRW+	1	zxxxxxaa
2	+ART +	1	das
3	+SUB +	1	Hauptfanggebiet
4	+PREP+	1	für
5	+SUB +	1	Schwertfische
6	+PREP+	1	im
7	+Z +	1	z+
8	+SUB +	1	Mittelmeer46
9	+Z +	1	+z
10	+VRB +STV +	1	liegt
11	+ADJ1+	1	eigentlich
12	+PREP+	1	in
13	+ART +	1	der
14	+Z +	1	z+
15	+SUB +	1	Straße
16	+PREP+	1	von
17	+SUB +	1	Messina09
18	+HS +AUES+	1	das Hauptfanggebiet für Schwertfische im z+ Mittelmeer46 +z liegt eigentlich in der z+ Straße von Messina09 +z .

SATZ-NUMMER: 2

da haben die Italiener eine sehr große Fischerei26 und auch eine sehr interessante Fischerei09 .

LFD.NR	MERKMALE	SQN	TEXTSEQUENZ
19	+ADV +	1	da
20	+VRB +SWV +	1	haben
21	+ART +	1	die
22	+SUB +	1	Italiener
23	+ART +	1	eine
24	+ADV +	1	sehr
25	+ADJ2+	1	große
26	+SUB +	1	Fischerei26
27	+KUNK+	1	und
28	+ADV +	1	auch
29	+ART +	1	eine
30	+ADV +	1	sehr
31	+ADJ2+	1	interessante
32	+SUB +	1	Fischerei09
33	+NG +LOK +	1	da
34	+VG +3P +PL +PRE +ID +	1	haben
35	+HS +AUES+	1	da haben die Italiener eine sehr große Fischerei26 und auch eine sehr interessante Fischerei09 .

Abbildung 24

Das Programm PC-50 kann auch dazu verwendet werden, eine Liste erfaßter Codierungen für den eigenen Gebrauch zu erstellen: es liefert alle gesammelten Codierungen mit den zugehörigen Textsequenzen.

Soll nur eine Auswahl, beschrieben durch Ebenenindizes, geliefert werden, steht das Programm PC-51: DECODIEREN UMGEWANDELTER CODIERUNGEN AUF AUSGEWÄHLTEN EBENEN zur Verfügung. Auch dieses Programm belegt jede Codierung mit ihrer zugehörigen Nummer und zwar bezogen auf den ganzen Bestand.

5.4 Umcodieren zu verändertem Codeumsetzer

Wird ein Codeumsetzer geändert, nachdem bereits Codierungen erstellt sind, ist es im allgemeinen notwendig, diese Codierungen anzupassen, damit sie schließlich durch den neuen Codeumsetzer interpretiert werden können.

Falls nur einige Merkmale hinzugefügt werden, kann es sein, daß sich die Veränderung des Codeumsetzers nicht auf die Interpretierbarkeit der Codierungen auswirkt, dann nämlich, wenn sich die Länge und damit die Inzidenz für die einzelnen Klassen nicht geändert hat. Die Länge bleibt erhalten, wenn auf der alten Anzahl benötigter Bits auch die Merkmale mit dem neu sich ergebenden Merkmalsumfang verschlüsselt werden können.

Beispiel: 5 Merkmale benötigen zur Verschlüsselung 3 Bits, erst für 8 Merkmale muß noch ein weiteres Bit hinzugezogen werden. Also können noch 2 Merkmale in die Klasse aufgenommen werden, ohne daß ein Umcodieren erforderlich ist.

Dazu aber muß auch die Belegung der Merkmale mit einem Wert, wenigstens für die bereits vorhandenen, erhalten bleiben. Das wird dadurch erreicht, daß neu aufzunehmende Merkmale in der zutreffenden Klasse auf die bereits vorhandenen aufgereiht werden und die Reihenfolge der vorhandenen unverändert bleibt.

Beispiel:

Codeumsetzer alt:	Merkmal	Wert	Länge	Inzidenz
	M1	1	3	10
	M2	2	3	10
	M3	3	3	10
	M4	4	3	10
	M5	5	3	10

Erweitern um die Merkmale N1 und N2 liefert:

Codeumsetzer neu:	Merkmal	Wert	Länge	Inzidenz
	M1	1	3	10
	M2	2	3	10
	M3	3	3	10
	M4	4	3	10
	M5	5	3	10
	N1	6	3	10
	N2	7	3	10

Dieser neue Codeumsetzer wird durch das Programm PC-12: VERÄNDERN EINES CODEUMSETZERS erstellt, das den alten Codeumsetzer in einer Sy-

stemdatei reserviert. Das Protokoll der neu erhaltenen Codierung wird dann mit der alten Codierung verglichen. Haben sich Länge und Inzidenz für die einzelnen Klassen nicht verändert und ist den alten Merkmalen noch der gleiche Wert zugeordnet, kann ein Umcodieren entfallen und die Systemdatei wird wieder freigegeben.

Andernfalls aber müssen alle Codierungen, die sich auf den alten Codeumsetzer beziehen, durch das Programm PC-60: UMCODIEREN FÜR NEUEN CODE-UMSETZER bearbeitet werden. Eingabedaten für dieses Programm sind alle vorhandenen STMFIL zu Texten. Diese Dateien werden in neue STMFIL copiert, wobei die einzelnen Codierungsangaben dem neuen Codeumsetzer angepaßt werden. Ein Umcodieren ist allerdings nur für die Ebenen erforderlich, in deren Merkmalsvorrat verändernd eingegriffen wurde. Diese Ebenen werden dem Programm mitgeteilt. Merkmale, die im alten Codeumsetzer enthalten waren, aber im neuen nicht mehr aufgeführt sind, werden nicht übernommen, so daß auf diese Weise auch das Löschen von Merkmalen möglich ist.

Will man von vornherein die Erweiterung eines Codeumsetzers in Betracht ziehen, empfiehlt es sich, Merkmale unter Pseudonamen in den Codeumsetzer mit aufzunehmen. Natürlich kann dies nur soweit geschehen, wie es die Kapazität des Codeumsetzers zuläßt. Sind dann im Rahmen der fortschreitenden Arbeit relevante Merkmale gefunden, wird der Codeumsetzer durch PC-12 neu erstellt, wobei an die Stelle der Pseudonamen die mnemonischen Kurzwörter der neu vereinbarten Merkmale treten, aus deren Anzahl jetzt keine Veränderung in Länge und Inzidenz einer Klasse resultiert. Demnach entfällt ein Umcodieren, ebenso kann die Systemdatei sofort freigegeben werden.

Beispiel: Im ersten Ansatz der Freiburger Codierungen auf Ebene 2 für die Klasse 6 (Kasusinformation) seien nur die Fälle

NOMINATIV
GENITIV
DATIV
AKKUSATIV

vorgesehen, eine Erweiterung um etwa zwei entsprechende Merkmale aber in Betracht gezogen. Also wird für den ersten Ansatz mit

NOM
GEN
DAT
AKK
DUM1
DUM2

ein Codeumsetzer so erstellt, daß in jedem Fall nachträglich geändert werden kann. Ein folgender Ansatz enthält dann schließlich die Merkmale, wie sie in Abbildung 3 aufgeführt sind.

In der Regel liegt, wenn der Einsatz der Parallelcodierung geplant ist, ein durchdachter Merkmalsvorrat zugrunde. Das Beispiel sollte lediglich zeigen, daß Parallelcodierung auch bereits auf einer Primärstufe durchgeführt werden kann, bei der das Erkennen aller Phänomene zugleich Arbeitsergebnis ist. Die Variabilität des Verfahrens unterstützt diese Arbeitsweise.

Auf diese Weise kann auch die Namensgebung für schon definierte Merkmale geändert werden, da

ein STMFIL zu einem Text nicht die Namen der Merkmale, sondern die interne Verschlüsselung enthält. Ein Decodieren des STMFIL liefert nach einer Umbenennung dann die Merkmale unter ihrem neuen Namen.

6. RETRIEVAL-VERFAHREN

Bis zu dieser Stelle wurde die Parallelcodierung als ein Verfahren zur Erfassung und Speicherung von Informationen zu Texten beschrieben. Sinnvollerweise muß der folgende Teil das Retrieval, also die Zugriffsmöglichkeiten auf die gespeicherten Daten, behandeln.

Es gibt eine Vielzahl linguistischer Fragestellungen, für deren Bearbeitung die Unterstützung durch ein solches Retrieval-System von Nutzen ist. Zu statistischen Untersuchungen (als Selbstzweck oder z. B. als Grundlage für didaktische Applikationen) liefert das System Daten über die Häufigkeit von Einzelphänomenen oder von Kombinationen, Daten über Korrelationen von Merkmalen und Daten zur Erforschung und Ermittlung von Kenngrößen zu Texten. Strukturanalysen syntaktischer oder semantischer Natur (von der Entwicklung von Wortstellungsregeln und Umgebungsanalysen bis hin zu komplexen inhaltsbezogenen Abhängigkeitsstrukturen) gehören ebenfalls zum Anwendungsspektrum des Retrieval-Systems mit Hilfe der Parallelcodierung.

6.1 Die Logik von Suchbegriffen

Die automatische Bearbeitung von Fragestellungen setzt die Möglichkeit des Zugriffs zu den relevanten Informationen voraus. Das System enthält in seiner derzeitigen Fassung zwei Programme, welche den parallelcodierten Informationspool und den zugehörigen Text für Fragestellungen öffnen.

Die Suchanfrage, die den Zugriff steuert, kann man sich allgemein so formuliert denken:

"Gesucht wird eine Textsequenz, die sich durch folgende Eigenschaften auszeichnet: ....."

Die geforderten Eigenschaften werden dabei mit Hilfe der mnemonischen Kurzwörter formuliert, setzen sich also aus einzelnen Merkmalsbeschreibungen zusammen, die zu einem "Suchbegriff" verbunden sind. Wir wollen an dieser Stelle die möglichen Verknüpfungen der Bausteine eines Suchbegriffs betrachten. Dazu verwenden wir die Begriffe der Aussagenlogik: und ($\wedge$), oder ($\vee$), nicht ($\neg$).

Im einfachsten Fall enthält ein Suchbegriff nur ein Merkmal a. Soll eine Suche Erfolg haben, wenn eine Textsequenz durch dieses Merkmal ausgezeichnet ist, verbinden wir a mit der Forderung "MIT" (selektives Suchverfahren), andernfalls mit der Forderung "OHNE" (restriktives Suchverfahren), wenn also nur die Textsequenzen zutreffen sollen, die nicht dieses Merkmal enthalten. Ein Suchbegriff, beispielsweise

KJ (MIT) führt auf Sequenzen, die im Konjunktiv stehen, und

KJ (OHNE) auf alle anderen.

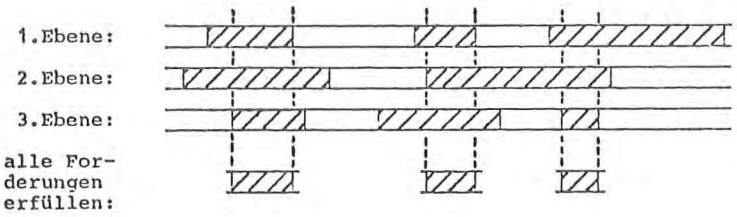
Das folgende Beispiel soll die Schreibweise darstellen und ihre Interpretation erklären:

a,b,...,x(MIT)
u,v,...,z(OHNE) $\rightarrow \{t; a \wedge b \wedge \ldots \wedge x \wedge \neg u \wedge \neg v \wedge \neg \ldots \wedge \neg z\}$ ($\neq$)

Die linke Seite spezifiziert dabei Merkmale a, b, ... , z, die beschrieben werden können durch mnemonische Kurzwörter, und ordnet ihnen eine Forderung "MIT" oder "OHNE" bei. Sie bildet den Suchbegriff, wie er ähnlich für ein Retrieval formuliert wird. Den Pfeil, der auf eine Menge verweist, verstehen wir dann als "führt auf" oder "liefert". Die rechte Seite beschreibt schließlich das Ergebnis des Suchvorgangs: die Menge aller Textsequenzen (oder Sätze) t, für die es Codierungen gibt, die zugleich (a∧b∧ ... ∧x) enthalten, nicht aber (u∧v∧ ... ∧z)

Um die möglichen Formen eines Suchbegriffs kennenzulernen und zu untersuchen, welche Implikationen diese nach sich ziehen, betrachten wir zuerst den einfachen Fall, daß nur drei Merkmale angegeben sind, die alle auf v e r s c h i e d e n e n E b e n e n liegen. Welche Forderungen mit den einzelnen Merkmalen verknüpft sind, ist an dieser Stelle gleichgültig. Es soll hier lediglich auf die Relationen zwischen den Ebenen ankommen.

Die Balken der nachfolgenden Figur mögen einen Text-Record bezeichnen, der mit irgendwelchen, den Text bildenden Zeichen ausgefüllt ist. Die Figur enthält dreimal den gleichen Record, jeweils von verschiedenen Ebenen aus betrachtet. Die schraffierten Stellen markieren dabei Textsequenzen, die durch Codierungen belegt sind, die das für die jeweilige Ebene angegebene Merkmal und seine Forderung erfüllen.



Ein solcher Record erfüllt dann alle Forderungen, wenn der Durchschnitt für alle auf den einzelnen Ebenen produzierten Textsequenzen nicht leer ist (ohne Einschränkung der Allgemeinheit seien die angegebenen Merkmale von der Forderung "MIT" begleitet):

a_1 (MIT) → $\{t; a_1\}$ für die erste Ebene

a_2 (MIT) → $\{t; a_2\}$ für die zweite Ebene

a_3 (MIT) → $\{t; a_3\}$ für die dritte Ebene

Und schließlich:

$$\{t;a_1\} \cap \{t;a_2\} \cap \{t;a_3\} = \{t;a_1 \wedge a_2 \wedge a_3\}$$

Die Forderungen zu mehreren Ebenen werden also durch das logische UND verknüpft.

Soll eine Menge, etwa $\{t; a_1 \vee a_2\}$ gefunden werden, die Forderungen also in wenigstens einem Falle erfüllt sein, kann aufgelöst werden nach:

$$\{t;a_1 \vee a_2\} = \{t;a_1\} \cup \{t;a_2\}$$

In dieser Form können die auf der rechten Seite enthaltenen Mengen in zwei getrennten Suchläufen gefunden werden, falls man nicht wie weiter unten beschrieben vorgehen will. (Es wäre problemlos, die zutreffenden Programme dahingehend zu erweitern, daß diese Form, oder auch

$$\{t;a_1 \vee a_2 \vee a_3\} = \{t;a_1\} \cup \{t;a_2\} \cup \{t;a_3\}$$

$$\{t;a_1 \vee a_2 \wedge a_3\} = \{t;a_1\} \cup \{t;a_2\} \cap \{t;a_3\}$$

$$\{t;a_1 \wedge a_2 \vee a_3\} = \{t;a_1\} \cap \{t;a_2\} \cup \{t;a_3\}$$

in einem Durchgang bearbeitet würden, allerdings reicht hierzu der Kernspeicherbereich auf der verwendeten Anlage nicht aus.)

Im zweiten Schritt betrachten wir die Produktionen eines Suchbegriffs für eine e i n z e l n e E b e n e . Dabei gehen wir zuerst davon aus, daß die aufgeführten Merkmale a, b, ... alle aus v e r s c h i e d e n e n K l a s s e n stammen, wobei es gleichgültig ist, ob es sich jeweils um ein obligatorisches oder ein fakultatives Merkmal handelt.

a (MIT) → {t; a} geliefert werden alle Textsequenzen, für die das Merkmal a zutrifft.

a, b, ... , x (MIT) → {t; a ∧ b ∧ ... ∧ x} liefert dann in Erweiterung alle Textsequenzen, für welche die angegebenen Merkmale zugleich zutreffen.

a (OHNE) → {t; ¬a}

a, b, ... , x (OHNE) → {t; ¬(a∧b∧ ... ∧x)} schließt das Zutreffen der angegebenen Merkmale aus, liefert demnach die Sequenzen, für die beliebig andere, nicht aber die gegebenen Merkmale zutreffen.

Eine Verknüpfung wie in (≠) ist damit als simultanes Zutreffen aller Forderungen erklärt.

Sind zwei Merkmale a und b nun E l e m e n t e d e r s e l b e n K l a s s e , ist ein Suchbegriff der Form

a, b (MIT)

sinnlos, da in einer Codierung nicht zwei Repräsentanten einer Klasse zugleich enthalten sein können. Die Form

a, b (OHNE)

dagegen ist durchaus sinnvoll. Durch sie werden nämlich lediglich bestimmte Merkmale ausgeschlossen, so daß andere Merkmale der Klasse durchaus zutreffen können.

Seien m_1, m_2, ... m_k alle Merkmale einer Klasse und n_1, n_2 ... n_i ($i<k$) eine Auswahl aus ihnen. Durch Umnumerieren erreichen wir $n_1=m_1$, $n_2=m_2$, usw. Dann gilt:

$$m_{i+1},m_{i+2},\ldots,m_k\,(\text{OHNE}) \rightarrow \{t;\ m_1 v m_2 v \ldots v m_i\}$$

Damit ist also auch die Adjunktion von Merkmalen möglich.

Ein erstes Beispiel aus den Freiburger Codierungen illustriert diesen Fall: Es sollen alle Verbalgruppen, die im Futur stehen, gesucht werden. Der Suchbegriff lautet dann:

VG (MIT)

PRE,IPF,PER,PQU (OHNE)

Die folgenden Suchbegriffe sind Beispiele zur Verdeutlichung des oben Gesagten. Sie bedienen sich ebenfalls der im Freiburger Codeumsetzer enthaltenen Merkmale.

a: EBENENAUSWAHL O1/O3

O1: MIT +KONS+TEMP+

O3: OHNE +SELE+

Geliefert werden Sätze mit subordinierender temporaler Konjunktion, die nicht in "semantisch leerem Tempus" stehen.

b: EBENENAUSWAHL O1/O2

O1: MIT +STVS+

O2: MIT +ID +

O2: OHNE +PRE +F1 +F2 +

Geliefert werden Sätze mit dem starken Verb "sein" im Indikativ Imperfekt, Perfekt oder Plusquamperfekt.

c: EBENENAUSWAHL O2/O3

O2: MIT +VG +1P +

O3: MIT +SELE+

Führt auf Sätze mit Verbalgruppen in der ersten Person, die in "semantisch leerem Tempus" stehen.

Neben Merkmalen, die einen Text begleiten, soll aber auch der gleichzeitige Zugriff auf den Text selbst möglich sein. Dazu kann in einem Suchbegriff zusätzlich eine Wortform angegeben werden, die dann ebenfalls die Forderung "MIT" oder "OHNE" enthalten kann, je nachdem ob eine über die Merkmalsangaben ausgesonderte Textsequenz

diese Wortform enthalten soll oder nicht.

Die Programme sehen auch eine Umsetzung der auf Lochkarten als dem Eingabemedium in Großschreibung spezifizierten Wortform in Kleinschreibung vor:

FISCH* → Fisch

MAN → man

Es ist erforderlich, hierbei auch die Anzahl der Zeichen der Wortform anzugeben, da ein auf die Wortform folgendes Leerzeichen, falls es berücksichtigt werden soll, zu unterschiedlichen Ergebnissen führen kann.

FISCH *(Länge 5) → Fisch

→ Fischfang

→ Fischerei

⋮

FISCH*(Länge 6) → Fisch_

So enthält ein Suchbegriff schließlich Merkmale und möglicherweise auch eine Wortform, die mit bestimmten Forderungen verknüpft sind. Die Auswahl der Merkmale, verbunden mit den gewünschten Forderungen, ihre Verteilung über maximal drei Ebenen und innerhalb der Ebenen über die einzelnen Merkmalsklassen erzeugt einen logischen Ausdruck. Dann können Sätze oder Sequenzen gesucht und gefunden werden, die diesem logischen Ausdruck genügen.

6.2 Satz-orientiertes Retrieval

Das Programm PC-70: SATZ RETRIEVAL liefert alle Sätze eines Textes, in denen sich ein vorgegebener Suchbegriff wenigstens einmal vollständig realisiert. Eine Kennzeichnung der zutreffenden Textsequenzen selbst ist dabei nicht möglich, ebenso erfolgt keine statistische Auswertung des Suchvorgangs.

Der Suchbegriff wird folgendermaßen formuliert (siehe dazu Abbildung 25):

1: Einleitung eines Suchbegriffs (EBENENAUSWAHL):

Für den Parameter EBENENAUSWAHL werden bis zu drei Zahlen angegeben, nämlich die Indizes der Codierungsebenen, auf denen das Retrieval stattfinden soll. Nur Merkmale dieser Ebenen können im folgenden gewählt werden.

2: Merkmalsauswahl:

Für jede gewählte Ebene kann ein obligatorisches Merkmal angegeben werden. Zusätzlich kann die Forderung "OHNE" mit ihm verbunden werden, die bedeutet, daß für die zugehörige Ebene alle anderen obligatorischen Merkmale zutreffen dürfen, nur das angegebene nicht. Fehlt die Angabe "OHNE", wird automatisch "MIT" substituiert. Für jede Ebene können weiter bis zu 10 fakultative Merkmale mit der Forderung "MIT", und ebenso bis zu 10 fakultative Merkmale mit der Forderung "OHNE" angegeben werden. Die Merkmale werden dabei nach ihrer Ebene, innerhalb der Ebene nach den Rubriken

- OBLIGATORISCH
- FAKULTATIV MIT
- FAKULTATIV OHNE

zusammengefaßt. Selbstverständlich müssen sie im Codeumsetzer enthalten sein. Es spielt lediglich die Zusammenfassung, nicht aber die Reihenfolge der einzelnen Merkmale innerhalb einer Rubrik oder die Aufeinanderfolge der Rubriken eine Rolle. Die Angaben in einer Rubrik werden auf einer Lochkarte mit dem zugehörigen Ebenenindex festgehalten. Wo keine Angabe vorhanden ist, entfällt diese Karte.

3: Wortform:

Angegeben werden kann eine Wortform aus maximal 10 Zeichen. Ist ein Stern (*) vorangestellt, wird das erste Zeichen der Wortform nicht verändert. Die folgenden Zeichen werden, ebenso auch das erste, falls kein Stern vorhanden, in Kleinschreibung umgesetzt nach folgender Vorschrift:

A ... Z → a ... z
% → ä
> → ö
? → ü
@ → Ä
= → Ö
" → Ü
$ → ß

Alle anderen Zeichen bleiben unverändert. Das Zeichen * vor der Wortform entfällt für @ (Ä), =(Ö) und "(Ü). Es kann eine Längenangabe gemacht werden. Fehlt sie, wird die vorliegende Anzahl der Zeichen als Länge genommen.

4: Beendigung eines Suchbegriffs:

Das Ende eines Suchbegriffs wird durch die Zei-

chenfolge "++++" markiert. Danach kann ein weiterer Suchbegriff für die Anwendung auf den gleichen Text folgen. Wieviele Suchläufe in dieser Weise nacheinander gestartet werden ist gleichgültig.

Ein Suchbegriff wird demnach eingeleitet durch den Parameter EBENENAUSWAHL und durch ++++ abgeschlossen. Die Anordnung der übrigen Daten dazwischen ist beliebig. Maximal können so 10 Rubriken (= Lochkarten) einschließlich einer gegebenen Wortform eingebettet werden. Abbildung 25 zeigt das Ablochschema in maximal ausgelegter Form für einen derartigen Suchbegriff.

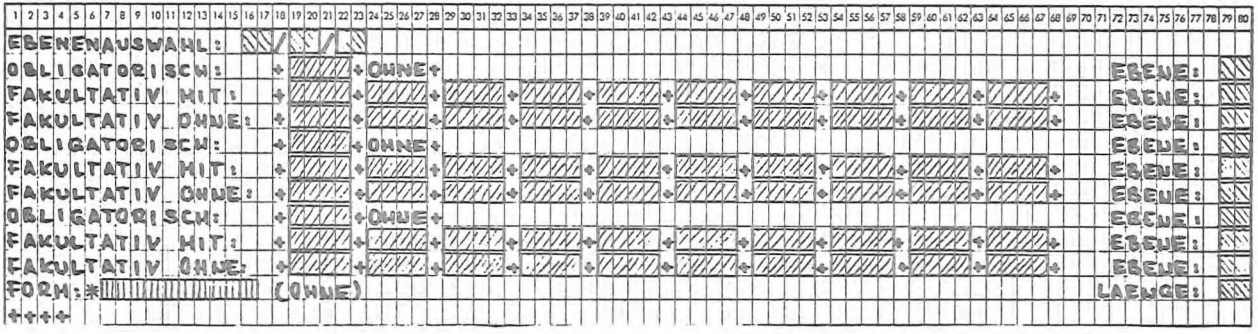
6.3 S t a t i s t i k - o r i e n t i e r - t e s R e t r i e v a l

Genau wie beim Satz-orientierten Retrieval wird auch hier ein Text über die zugehörigen Codierungen anhand eines formulierten Suchbegriffs durchsucht. Dabei kann allerdings über Optionen angegeben werden, welche Text-Ausgabe geliefert werden soll: zutreffende Sätze, zutreffende Sequenzen oder gar keine Text-Ausgabe. In diesem Programm, PC-71: STATISTIK-ORIENTIERTES RETRIEVAL, werden Zählungen vorgenommen. Eine Zählung kann sich dabei auf die zutreffenden Sätze beziehen, und über die Anzahl der durchsuchten Sätze kann die relative Häufigkeit (in Prozent) der Realisationen des Suchbegriffs zum Text berechnet werden. Eine Zählung ist weiter möglich für die Anzahl der Realisationen der Merkmale auf e i n e r Ebene im Suchbegriff. Die Codierungsebene, auf der diese Zählung stattfinden soll, wird als Zählebene bezeichnet. Es ist gewöhnlich die Ebene, auf der das Hauptgewicht der Untersuchung liegt, wobei andere Ebenen nur zur weiteren Abgrenzung herangezogen werden. Eine Zählung auf Zählebene liefert die Anzahl

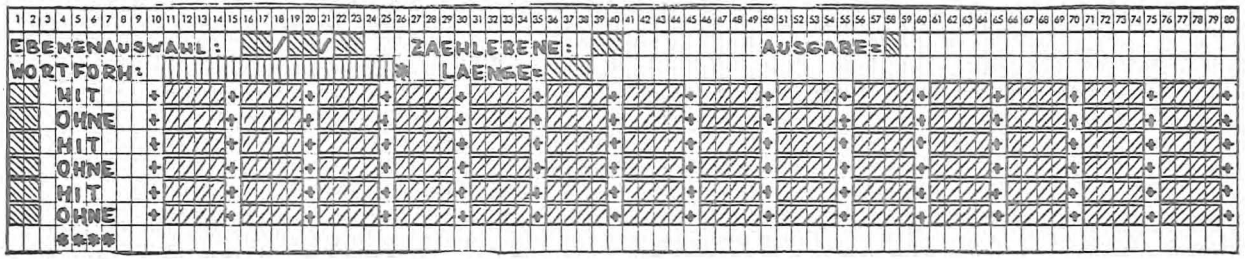
Abbildung 25

CODIERUNG VON SUCHBEGRIFFEN

PC-70: SATZ-ORIENTIERTES RETRIEVAL



PC-71: STATISTIK-ORIENTIERTES RETRIEVAL



= EBENENINDEX ODER SONSTIGER NUMERISCHER EINTRAG (RECHTSBUENDIG)

= MNEMONISCHES KURZWORT EINES MERKMALS (LINKSBUENDIG)

= BELIEBIGE ZEICHEN, DIE DEN UMWANDLUNGSVORSCHRIFTEN UNTERWORFEN SIND (LINKSBUENDIG)

der untersuchten Codierungen auf dieser Ebene, sowie die Anzahl der Erfolge, und zwar für jeden Satz einzeln, wie auch für den ganzen Text.

Ein Suchbegriff mit den gewünschten Optionen wird folgendermaßen formuliert (siehe dazu Abbildung 25):

1: Einleitung eines Suchbegriffs:

a: EBENENAUSWAHL: Hier wird, wie auch bei PC-70, angegeben, zu welchen Ebenen Merkmale folgen. Es sind wiederum maximal drei Ebenen zugleich spezifizierbar, und zwar durch ihre Ebenenindizes.

b: ZAEHLEBENE: Sie gibt den Index der Ebene an, auf der eine Zählung nach durchsuchten und zutreffenden Codierungen vorgenommen werden soll. Dieser Index darf nicht fehlen und muß auch unter den bei EBENENAUSWAHL aufgeführten enthalten sein.

c: AUSGABE: Hier können vier verschiedene Formen eines Retrieval-Protokolls gewählt werden.

AUSGABE=0 Es wird lediglich der erfaßte Suchbegriff, sowie eine Statistik des Suchlaufs nach seiner Beendigung gegeben.

AUSGABE=1 Es wird die vollständige Textausgabe gegeben, und zwar

- Protokoll des Suchbegriffs
- alle Sätze, in denen sich der Suchbegriff wenigstens einmal realisiert
- zu jedem Satz das Ergebnis der Zählung auf Zählebene, und zwar mit Satznummer, Anzahl der Codierungen auf Zählebene, sowie An-

zahl der Erfolge unter ihnen

- Statistik des Suchlaufs nach Beendigung des Retrieval zu dem Suchbegriff.

AUSGABE=2 liefert die gleichen Protokollteile wie AUSGABE=1, allerdings wird nicht der gesamte Textsatz abgedruckt, sondern lediglich jede zutreffende Sequenz.

AUSGABE=3 Diese Option schließlich liefert nur statistische Angaben: Zählergebnisse auf Zählebene für zutreffende Sätze und Gesamtstatistik des Durchlaufs.

Unter der Statistik des Suchlaufs wird tabellarisch zusammengefaßt:

- die Gesamtanzahl der durchsuchten Sätze, die Anzahl der zutreffenden Sätze, in denen sich der Suchbegriff also realisiert, und der daraus resultierende Prozentsatz des Erfolgs.

- die Gesamtanzahl aller auf Zählebene vorgelegenen Codierungen, sowie die Anzahl unter ihnen, auf welche die für diese Ebene angegebenen Merkmale zutrafen, und schließlich der daraus resultierende Prozentsatz des Erfolgs.

2: Wortform:

Auf einer Lochkarte kann wieder eine Wortform angegeben werden, die in einer zutreffenden Textsequenz enthalten sein soll bzw. nicht sein darf. Soll der Suchbegriff keine Wortform enthalten, bleibt diese Karte leer, muß aber vorhanden sein. Eine Wortform kann maximal 15 Zeichen lang sein, die Zeichen werden wieder in Kleinschreibung um-

gesetzt nach der gleichen Vorschrift, die auch für das Satz-orientierte Retrieval durch PC-70 gilt (siehe Seite 109). Wird nach der Wortform das Zeichen "*" angegeben, wird das erste Zeichen in Großschreibung beibehalten. * entfällt wieder für @ (Ä), =(Ö) und "(Ü). Alle anderen, nicht in der Abbildungsvorschrift aufgeführten Zeichen werden unverändert übernommen. Für die Wortform muß eine Längenangabe gemacht werden. Positive Länge zeigt an, daß die Forderung "MIT", negative Länge, daß die Forderung "OHNE" gelten soll. Wie auch beim Satz-orientierten Retrieval kann durch die Längenangabe ein nachfolgendes Leerzeichen an die Wortform gebunden werden.

3: Merkmalsauswahl:

Für diese Form des Retrieval wird keine Unterscheidung nach obligatorischen und fakultativen Merkmalen vorgenommen. So können Merkmale beliebig in den einzelnen Rubriken zusammengefaßt sein. Für jede unter EBENENAUSWAHL bezeichnete Ebene können zwei Rubriken auftreten: Merkmale mit der Forderung "MIT" und Merkmale mit der Forderung "OHNE". Da maximal drei Ebenen zulässig sind, können so bis zu 6 Rubriken in einem Suchbegriff enthalten sein. Eine Rubrik faßt die Merkmale einer Ebene zusammen, welche alle die gleiche Forderung stellen, und zwar bis zu 14. Die Angaben zu einer Rubrik werden formuliert durch

- Ebenenindex
- MIT oder OHNE
- gewünschte Merkmale

Auch hier spielt die Anordnung der Merkmale in einer Rubrik, sowie die Aufeinanderfolge der Rubriken keine Rolle. Wo keine Angaben gemacht werden, wird auch keine Lochkarte erstellt.

4: Beendigung eines Suchbegriffs:

Das Ende eines Suchbegriffs markiert die Zeichenfolge "* * * *". Danach können weitere Suchbegriffe zum gleichen Text in beliebiger Anzahl folgen. Das Ablochschema in maximal ausgelegter Form ist ebenfalls in Abbildung 25 enthalten.

Die folgenden Abbildungen zeigen die Ergebnisse eines Suchlaufs über einem Text. Es wurden dabei gesucht: Nominalgruppen im Singular, die nicht Präpositionalphrase sind, und ein Substantiv, hier die Wortform "Fisch" enthalten.

Abbildung 26 zeigt den Maschinenausdruck, wie er für alle Ausgabeformen gegeben wird. Diese Form erfährt keine Erweiterung für AUSGABE=O.

Die Abbildungen 27 und 29 zeigen dann die zusätzliche Ausgabe für die Parameterwerte 1, 2 und 3.

Wäre im Suchbegriff als Länge der Wortform der Wert 6 angegeben, so wäre der Satz 53 allein zutreffend. Es liegt am Charakter der Freiburger Quelltexte, die zusätzlich Intonationszeichen enthalten, daß die Sätze 50 und 51 in diesem Fall nicht geliefert werden.

EBENENAUSWAHL: 2/ 1/ 0/

PARAMETER: ZE= 2 AU=0

PROTOKOLL DES SUCHBEGRIFFS (PC-71)

(fuer alle Formen der Ausgabe)

ZE = ZAEHLEBENE
AU = AUSGABE

WORTFORM: Fisch (5)

EBENE 2(MIT)	+NG	+SI	+	+	+	+	+	+	+	+	+	+	+	+	+
EBENE 2(OHNE)	+PP	+	+	+	+	+	+	+	+	+	+	+	+	+	+
EBENE 1(MIT)	+SUB	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Abbildung 26

STATISTIK DES SUCHLAUFS

(PC-71)

STATISTIK DES SUCHLAUFS **********	GESAMT	ERFOLGE	PROZENT
DURCHSUCHTE SAETZE	85	7	8.24
CODIERUNGEN AUF ZAEHLEBENE	389	7	1.80

TEXTPROTOKOLL FUER AUSGABE = 1

SATZ-NUMMER: 4

+p+ und3 diese z+ Malta4-Fischersi +z auf Schwertfische5 is eigentlich eine Zufallsfischerei09 .

CODIERUNGEN AUF ZAEHLEBENE: 2 , DARAUS 1 ERFOLG(E) IN SATZ 4

SATZ-NUMMER: 33

/ haben mir erzählt27 s+ also die letzte Fischerei wäre grauslich gewesen26 +s hätten so gut wie nichts bekommen29 und wir haben nun also angefangen7 i+ zu holen19 +i .

CODIERUNGEN AUF ZAEHLEBENE: 3 , DARAUS 1 ERFOLG(E) IN SATZ 33

SATZ-NUMMER: 50
---
man läßt also den Fisch4 sich austoben26 versucht26 ,+ sobald er seine5 Kreise4 zieht36 etwas näher an s Boot kommt26 +, i+ die Leine weiter hineinzunehmen26 +i i+ um ihn dadurch langsam an die Oberfläche zu bekommen16 +i .

CODIERUNGEN AUF ZAEHLEBENE: 5 , DARAUS 1 ERFOLG(E) IN SATZ 50

SATZ-NUMMER: 51

,+ und wenn er dann ziemlich nahe an der Oberfläche ist46 und schon etwas ermüdet ist26 +, versucht man57 i+ ihm eine Schlinge47 über den Schwanz zu ziehen26 +i i+ und4 mit Hilfe einer Winde4 den Fisch4 herauszuholen09 +i .

CODIERUNGEN AUF ZAEHLEBENE: 5 , DARAUS 1 ERFOLG(E) IN SATZ 51

SATZ-NUMMER: 53
---
da müssen also alle Mann ran26 i+ um den Fisch rüberzuholen26 +i .

CODIERUNGEN AUF ZAEHLEBENE: 1 , DARAUS 1 ERFOLG(E) IN SATZ 53

SATZ-NUMMER: 77

/ sondern ich war sehr froh47 ,+ daß ich das5 hatte sehen können27 +, denn is doch eine ziemlich seltene Fischerei26 ,+ wenn man auch vielleicht sagen kann26 +, (na ja) 5 also wollen mal sagen57 s+ unsere Fischer würden vielleicht sagen47 +s s+ das war n bißchen f+ Pütschkram26 +f +s .

CODIERUNGEN AUF ZAEHLEBENE: 7 , DARAUS 1 ERFOLG(E) IN SATZ 77

SATZ-NUMMER: 81
---
denn die Malteser fangen27 (seit neunzehnhundertvierundsechzig ist diese Fischerei erst47) pro Jahr4 etwa4 fünfzig5 Tonnen06 .

CODIERUNGEN AUF ZAEHLEBENE: 1 , DARAUS 1 ERFOLG(E) IN SATZ 81

Abbildung 28

```
TEXTPROTOKOLL FUER AUSGABE = 2


SATZ-NUMMER:      4
------------------
# 1  diese z+ Malta4-Fischerei +z
CODIERUNGEN AUF ZAEHLEBENE:   2 , DARAUS  1 ERFOLG(E) IN SATZ     4

SATZ-NUMMER:     33
------------------
# 1  die letzte Fischerei
CODIERUNGEN AUF ZAEHLEBENE:   3 , DARAUS  1 ERFOLG(E) IN SATZ    33

SATZ-NUMMER:     50
------------------
# 1  den Fisch4
CODIERUNGEN AUF ZAEHLEBENE:   5 , DARAUS  1 ERFOLG(E) IN SATZ    50

SATZ-NUMMER:     51
------------------
# 1  den Fisch4
CODIERUNGEN AUF ZAEHLEBENE:   5 , DARAUS  1 ERFOLG(E) IN SATZ    51

SATZ-NUMMER:     53
------------------
# 1  den Fisch
CODIERUNGEN AUF ZAEHLEBENE:   1 , DARAUS  1 ERFOLG(E) IN SATZ    53

SATZ-NUMMER:     77
------------------
# 1  eine ziemlich seltene Fischerei26
CODIERUNGEN AUF ZAEHLEBENE:   7 , DARAUS  1 ERFOLG(E) IN SATZ    77

SATZ-NUMMER:     81
------------------
# 1  diese Fischerei
CODIERUNGEN AUF ZAEHLEBENE:   1 , DARAUS  1 ERFOLG(E) IN SATZ    81
```

Abbildung 29

PROTOKOLL FUER AUSGABE = 3

```
CODIERUNGEN AUF ZAEHLEBENE:   2 , DARAUS  1 ERFOLG(E) IN SATZ     4
CODIERUNGEN AUF ZAEHLEBENE:   3 , DARAUS  1 ERFOLG(F) IN SATZ    33
CODIERUNGEN AUF ZAEHLEBENE:   5 , DARAUS  1 ERFOLG(E) IN SATZ    50
CODIERUNGEN AUF ZAEHLEBENE:   5 , DARAUS  1 ERFOLG(E) IN SATZ    51
CODIERUNGEN AUF ZAEHLEBENE:   1 , DARAUS  1 ERFOLG(E) IN SATZ    53
CODIERUNGEN AUF ZAEHLEBENE:   7 , DARAUS  1 ERFOLG(E) IN SATZ    77
CODIERUNGEN AUF ZAEHLEBENE:   1 , DARAUS  1 ERFOLG(E) IN SATZ    81
```

7. DATENSICHTUNG

Kehren wir zurück zu der Strukturbaumdarstellung auf Seite 107. Eine derartige Darstellung ist nützlich und wertvoll, zeigt sie doch auf einen Blick alle Zusammenhänge, die innerhalb des Satzes für die einzelnen Ebenen wirksam sind.

Wenn parallelcodierte Daten für einen Text vorhanden sind, soll es auch die Möglichkeit geben, diese Zusammenhänge durch die Maschine sich aufzeichnen zu lassen. Hier aber setzen nun technische Schwierigkeiten gewisse Grenzen.

Ein Schnelldrucker, der den Output für Text und Codierungen liefern kann, kann nicht zugleich die verbindenden Striche ziehen, über die der einzelne Bezug verfolgt wird, insbesondere dann nicht, wenn die zur Verfügung stehende Breite der Papierbahn eine nicht zu überschreitende Begrenzung setzt.

Das Programm PC-80: STRUKTURZEIGENDE TEXTAUFBEREITUNG erzeugt nun zu einem Text und den zugehörigen Codierungen einen Ausdruck, welcher der Strukturbaumdarstellung angenähert ist. Die Abbildungen 30 und 31 zeigen zwei Beispiele eines derartigen Ausdrucks. Die linke Spalte enthält dabei den Text, wortweise untereinander aufgelistet. Inhalt und Reihenfolge der restlichen drei Spalten ist beliebig wählbar. Sie liefern die Codierungen der einzelnen Sätze auf bis zu drei Ebenen für eine angebbare Ebenenauswahl. Sämtliche Codierungen zum Text auf diesen ausgewählten Codierungsebenen werden dann decodiert, wobei zusätzlich angegeben werden kann, welche Klassen von fakultativen Merkmalen berücksichtigt werden sollen (maximal 5 je ausgewählter

SATZ-NUMMER: 12

TEXT	EBENE 1 SON MERKMALE	EBENE 2 SON MERKMALE	EBENE 3 SON MERKMALE
und	1 +KONK+		1 +NS +AUES+
,+			2 +NS +EST +AUES+
damit	2 +KONS+		2 +
man	3 +PRON+	1 +NG +SI +	2 +
auch	4 +ADV +		2 +
in	5 +PREP+	2 +NG +SI +	2 +
der	6 +ART +	2 +	2 +
Dunkelheit56	7 +SUB +	2 +	2 +
diese	8 +ART +	3 +NG +SI +	2 +
treibende	9 +PTZ1+STV +	3 +	2 +
Leine	10 +SUB +	3 +	2 +
sieht26	11 +VRB +STV +	5 +VG +3P +SI +PRE +ID +	2 +
+,			2 +
kommt	12 +VRB +STV +	6 +VG +3P +SI +PRE +AK +ID +	1 +NS +AUES+
ab	13 +ADV +	7 +NG +	1 +
und	14 +KONK+	7 +	1 +
zu	15 +ADV +	7 +	1 +
ne	16 +ART +	8 +NG +SI +	1 +
Korkplatte5	17 +SUB +	8 +	1 +
mit	18 +PREP+	9 +NG +SI +	1 +
ner	19 +ART +	9 +	1 +
Petroleumlampe	20 +SUB +	9 +	1 +
drauf46	21 +ADV +	10 +NG +	1 +
,+			3 +NS +EST +AUES+
die	22 +ART +	11 +NG +SI +	3 +
also	23 +ZERO+		3 +
ein	24 +ART +	12 +NG +SI +	3 +
offenes	25 +ADJ2+	12 +	3 +
Feuer	26 +SUB +	12 +	3 +
hat26	27 +VRB +SWV +	13 +VG +3P +SI +PRE +ID +	3 +
+,			3 +
,+			4 +NS +EST +AUES+
so	28 +KONS+		4 +
daß	29 +KONS+		4 +
man	30 +PRON+	14 +NG +SI +	4 +
ganz	31 +ADJ1+		4 +
weit4	32 +ADJ1+		4 +
schon	33 +ADV +	15 +....+	4 +
diese4	34 +ART +	16 +NG +PL +	4 +
treibenden	35 +PTZ1+STV +	16 +	4 +
Langleinen3	36 +SUB +	16 +	4 +
erkennen	37 +INF +SWV +	17 +VG +3P +SI +PRE +ID +	4 +
kann09	38 +VRB1+SWV +	17 +	4 +
+,			4 +
.			

Abbildung 30

PARALLELCODIERUNG VON xaa SEITE: 63

SATZ-NUMMER: 64

TEXT	EBENE 1 SQN	MERKMALE	EBENE 2 SQN	MERKMALE	EBENE 3 SQN	MERKMALE
/					1	+SS +
und	1	+KONK+			2	+MS +AVS +
schließlich	2	+ADV +			2	+
gegen	3	+PREP+	1	+NG +SI +	2	+
Morgen26	4	+SUB +	1	+	2	+
(					3	+PT +AVS +
es	5	+PERS+	2	+----+	3	+
war	6	+VRB +STVS+	3	+VG +3P +SI +IPF +AK +ID +	3	+
oder	7	+PREP+LE +			1	+
beinah	8	+ADV +			3	+PT +AVS +
schon	9	+ADV +			3	+
Vormittag	10	+SUB +	4	+NG +SI +	3	+
)26					3	+
(					4	+PT +AVS +
die	11	+ART +	5	+NG +SI +	4	+
Sonne	12	+SUB +	5	+	4	+
war	13	+VRB +STVS+	6	+VG +3P +SI +IPF +AK +ID +	4	+
längst	14	+ADV +	7	+NG +	4	+
+g+					4	+
am	15	+PREP+	8	+NG +SI +	4	+
Himmel26	16	+SUB +	8	+	4	+
)					4	+
(					5	+PT +AVS +
und	17	+KONK+			5	+
es	18	+PERS+	9	+NG +SI +	5	+
war	19	+VRBZ+STVS+	10	+VG +3P +SI +PQU +AK +ID +	5	+
also	20	+ADV +			5	+
schon	21	+ADV +	11	+NG +	5	+
ziemlich	22	+ADJ1+			5	+
heiß	23	+ADJ1+			5	+
geworden47	24	+PTZ2+	10	+VG +3P +SI +PQU +AK +ID +	5	+
auf	25	+PREP+	12	+NG +	5	+
See26	26	+SUB +	12	+	5	+
)					5	+
kam5	27	+VRB +STV +			2	+MS +AVS +
ein	28	+ART +LE +			2	+MS +AVS +
Nummer	29	+SUB +	13	+NG +SI +	2	+
sechs26	30	+ADV +	13	+	2	+
.					1	+

Abbildung 31

Ebene; das obligatorische Merkmal einer Codierung wird prinzipiell geliefert). Das Ergebnis der Merkmalsrückfindung wird dann als Folge mnemonischer Kurzwörter in der für die Ebene zutreffenden Spalte auf der Zeile angegeben, auf der die zugehörige Textsequenz beginnt.

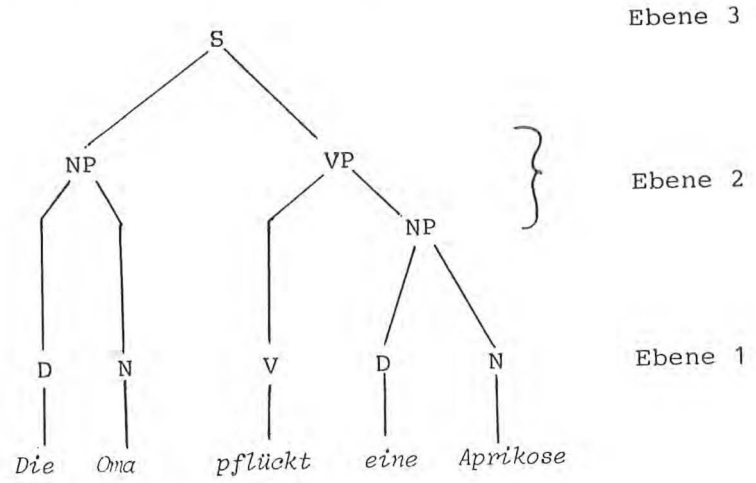
Zur eindeutigen Zuordnung erhält jede Codierung innerhalb ihrer Ebene eine Folgenummer (SQN) zugewiesen, die dann die vollständige Textsequenz begleitet, falls sie aus mehr als einem Wort besteht. Ist eine Textsequenz in mehrere Teile aufgespalten, wird für jeden einzelnen Teil unter gleicher Folgenummer nochmals die Reihe der mnemonischen Kurzwörter abgedruckt. Wird also für den Beispielsatz von Seite 45 die Ebenenauswahl 1 / 2 / 3 in dieser Reihenfolge getroffen, ergäbe sich folgendes Bild:

Text	Ebene 1	Ebene 2	Ebene 3
	Merkmale	Merkmale	Merkmale
Wir	1 +PERS+	1 +NG+1P+PL+NOM+	1 +HS+AVS+
sind	2 +VRB2+STVS+	2 +VG+1P+PL+PER+ID+AK+	1 +
dann	3 +ADV+	3 +NG+TEMP+	1 +
in	4 +PREP+	4 +NG+SI+PP+LOK+	1 +
das	5 +ART+	4 +	1 +
Brücken-haus	6 +SUB+	4 +	1 +
gekro-chen	7 +PTZ2+STV+	2 +VG+1P+PL+PER+ID+AK+	1 +

Eine andere Ebenenauswahl, etwa 3 / 2 / 1, würde eine Umordnung der Spalten nach sich ziehen. In gleicher Weise lassen sich durch eine beschränkende Klassenauswahl Merkmale unterdrücken, die

für die beabsichtigte Untersuchung nicht relevant sind.

Eine weitere Beschränkung muß hierbei in Kauf genommen werden. Betrachten wir folgende vereinfachte Darstellung (entnommen aus GROSS - LENTIN: Mathematische Linguistik):



Hier begleiten zwei Codierungen auf gleicher Ebene (Ebene 2) Teile ein und derselben Textsequenz. Dieses Problem kann von PC-80 nur dadurch gelöst werden, daß die zuletzt ankommende Information die vorherige überschreibt. So erhalten wir

	SQN		SQN		SQN	
Die	1	D	1	NP	1	S
Oma	2	N	1		1	
pflückt	3	V	2	VP	1	
eine	4	D	3	NP	1	
Aprikose	5	N	3		1	

Daß dabei Information tatsächlich überschrieben wird, ist aus Abbildung 31 für die Ebene 3 ersichtlich. Die Codierung +SS + bezieht sich auf den ganzen Satz. Mit der Folgenummer 1 wird sie auch für den ganzen Satz eingetragen. Nachfolgende Codierungen auf der gleichen Ebene überschreiben Teile davon mit ihrer Folgenummer; an einigen Stellen bleibt die 1 noch sichtbar. Daß es sich dabei um ein Überschreiben handelt, kann allgemein daraus erkannt werden, daß die Folge der mnemonischen Kurzwörter mit Beginn einer neuen Folgenummer nicht mehr erscheint.

ANHANG I

Anleitung zum Codieren und Ablochen

1. Codieren mit der Textvorlage aus PC-20/21
2. Das Ablochen aus der Codierungsvorlage
3. Codieren mit der Textvorlage aus PC-22
4. Die Bearbeitung des Ergebnisprotokolls von PC-30

1. CODIEREN MIT DER TEXTVORLAGE AUS PC-20/21

Ein zu codierender, satzzerlegter Text wird mit dem Programm PC-20 oder PC-21 als Ganzes oder in einzelnen Teilen als Codierungsunterlage zur Parallelcodierung formularmäßig ausgedruckt (siehe Abbildung 6). In die Vorlage können die Codierungen für bis zu vier Ebenen handschriftlich eingetragen werden. Aus der Vorlage kann direkt abgelocht werden, ohne daß ein feldgerechtes Umschreiben auf ein Ablochschema für Daten notwendig ist.

Die mit "ABLOCHUNTERLAGEN ZUR PARALLELCODIERUNG ... TEXTSCHLÜSSEL xyz ... SEITE nnn" überschriebenen Vorlagen gliedern sich in 6 Spalten: Die erste Spalte enthält fortlaufend untereinander die Textwörter eines Satzes, wie sie durch die Einschliessung in Leerstellen erkannt werden. Spalte 2 numeriert diese Wörter innerhalb des Satzes. Die Spalten 3 bis 6 dienen der handschriftlichen Eintragung der zutreffenden Codierungen auf den gewünschten Ebenen. Mit Beginn eines neuen Textsatzes wird über den 6 Spalten eine Kopfzeile gedruckt, die diese Aufteilung beschreibt und zusätzlich abzulochende Angaben enthält. Die in der Kopfzeile enthaltenen Daten umfassen:

* Textschlüssel	(3 Zeichen)	bereits vorgedruckt
* Satznummer	(5-stellig)	
* Ebenenindex	(2-stellig)	für jede Ebene anzugeben
* Codeformat	(1-stellig)	

Die Vorlage stellt auf jeder Ebene zu jedem Wort ein Kästchen bereit, das zur Aufnahme der handschriftlichen Codierungen dient. Wo keine Codierung zutrifft, bleibt dieses Kästchen leer. Eben-

so wird nur ein Kästchen ausgefüllt, wenn eine Codierung sich auf mehrere Wörter zugleich bezieht.

Eine Codierung setzt sich zusammen aus

- Codierungsangaben
- Positionsangaben.

Die Codierungsangaben sind die Merkmalsträger und bestehen aus Folgen von mnemonischen Kurzwörtern, die den zu codierenden Sachverhalt beschreiben und auf der zugehörigen Codierungsebene erklärt sind. Die Positionsangaben stellen den Bezug der Codierungsangaben zum Text her und beschreiben die zutreffende Textsequenz.

Ein Kästchen der Vorlage wird also mit zwei Zeilen beschriftet: die erste enthält die einzelnen mnemonischen Kurzwörter, die durch Pluszeichen "+" voneinander getrennt werden, die zweite enthält Zahlenangaben, die Positionsangaben, wobei die einzelnen Zahlen durch Kommata "," voneinander getrennt werden (siehe Abbildungen 9 und 10).

Die Positionsangaben werden charakterisiert durch das Codeformat, das darüber entscheidet, ob die zugehörige Textsequenz eine Zeichenkette (Wortbestandteil) ist oder ob sie durch vollständige Textwörter angebbar ist. Entsprechend wird in die Kopfzeile zur Codierungsebene eingetragen:

CODEFORMAT: 0 für Wortketten
CODEFORMAT: 1 für Zeichenketten

Das Codeformat braucht innerhalb einer Codierungs-

ebene nicht konstant beibehalten zu werden. Wird es für eine Codierung geändert, soll diese Änderung deutlich in dem zutreffenden Kästchen zusätzlich eingetragen werden, weil dann für diese Angaben nicht schematisch abgelocht werden kann (siehe dazu Seite 146 oben.

a: Positionsangaben für Wortketten (Codeformat O)

Prinzipiell können vier Fälle unterschieden werden: Die Textsequenz besteht aus

- genau einem Wort
- einer zusammenhängenden Folge von Wörtern
- einer nicht zusammenhängenden Folge von Wörtern
- dem ganzen Satz.

Besteht die Textsequenz aus genau e i n e m W o r t , wird als Positionsangabe der in Spalte 2 der Vorlage abgedruckte Wortindex angegeben, also eine einzige Zahl.

Für eine z u s a m m e n h ä n g e n d e F o l g e von Wörtern werden genau zwei Zahlen als Positionsangaben geschrieben, die durch Komma voneinander getrennt sind. Diese beiden Zahlen geben den Index des ersten und letzten Wortes der Textsequenz an, wie er aus Spalte 2 entnommen werden kann.

Die Positionsangaben für eine n i c h t z u s a m m e n h ä n g e n d e F o l g e von Wörtern aus beispielsweise n zusammenhängenden Gliedern bestehen aus genau n Paaren von Zahlen, wobei jedes Paar einen zusammenhängenden Bestandteil der Wortkette beschreibt, und zwar wieder

durch den Index des ersten und letzten Wortes. Besteht ein Glied einer nicht zusammenhängenden Wortkette aus nur einem Wort, so muß für diesen Teil aus einem Wort der gleiche Wortindex zweimal erscheinen, damit die Angaben insgesamt paarig bleiben und die Eindeutigkeit gewährleistet ist. Die einzelnen Zahlenangaben für die Wortindizes werden wieder durch Kommata voneinander getrennt. Aus programmtechnischen Gründen sind höchstens 6-gliedrige Wortketten zulässig.

Bezieht sich die Codierungsangabe auf den g a n z e n S a t z , brauchen keine Positionsangaben gegeben zu werden.

b: Positionsangaben für Zeichenketten (Codeformat 1)

Die Positionsangaben für Zeichenketten bestehen grundsätzlich aus Zahlentripeln. Der erste Wert in einem Tripel gibt den Wortindex an, bei dem beginnend die Zeichenkette gezählt wird. Der zweite Wert bezeichnet den Index des ersten Zeichens der Kette, vom Anfang des angegebenen Wortes aus gezählt (Beginn der Zeichenkette), der dritte Wert schließlich bezeichnet den Index des letzten Zeichens der Kette, ebenfalls vom Anfang des angegebenen Wortes aus gezählt.

Ein solches Tripel bezeichnet somit genau eine zusammenhängende Zeichenkette, deren Grenzen auf den Beginn eines Wortes im Satz bezogen sind. Besteht eine Zeichenkette aus nur einem Zeichen, so sind die letzten beiden Zahlenangaben im Tripel identisch, müssen aber vorhanden sein. Setzt sich die zu bezeichnende Textsequenz aus mehreren, nicht zusammenhängenden Zeichenketten zusammen, so bestehen die Positionsangaben aus so vielen Tripeln, wie die Zeichenkette Glieder enthält.

Die Gesamtanzahl k der in den Positionsangaben gegebenen Zahlen, die durch Kommata voneinander getrennt sind, ist somit in jedem Fall durch 3 teilbar. Aus programmtechnischen Gründen sind höchstens 4-gliedrige Zeichenketten zulässig; die Positionsangaben bestehen also aus höchstens 12 Zahlen.

2. DAS ABLOCHEN AUS DER CODIERUNGSVORLAGE

Prinzipiell gilt: Eine Codierung wird auf genau eine Lochkarte geschrieben. Wenn keine Codierung vorhanden ist, braucht auch keine Lochkarte erstellt zu werden.

Entsprechend der Datenunterscheidung in

- Textschlüssel
- Satznummer
- Ebenenindex
- Codeformat
- Codierungsangaben
- Positionsangaben

wird die Lochkarte in 6 Felder aufgeteilt (siehe Abbildung 32):

Spalten	1 - 3	Textschlüssel (in Großschreibung)
	4 - 8	Satznummer (wie in Codierungsvorlage)
	9 - 10	Ebenenindex (rechtsbündig)
	11	Codeformat

12 - 47 Codierungsangaben

48 - 80 Positionsangaben

In den Spalten 1 bis 11 muß spaltengerecht gelocht werden. Die Codierungs- und Positionsangaben werden spaltenfrei, ohne Rücksicht auf führende, eingebettete oder nachfolgende Leerzeichen abgelocht. Es ist lediglich darauf zu achten, daß sie in den vorgesehenen Feldern stehen. Als Trennsymbol zwischen zwei mnemonischen Kurzwörtern steht das Pluszeichen "+", Zahlen der Positionsangaben sind durch Komma "," voneinander zu trennen. Trennsymbole stehen nur zwischen zwei Angaben, nicht also am Anfang oder am Ende.

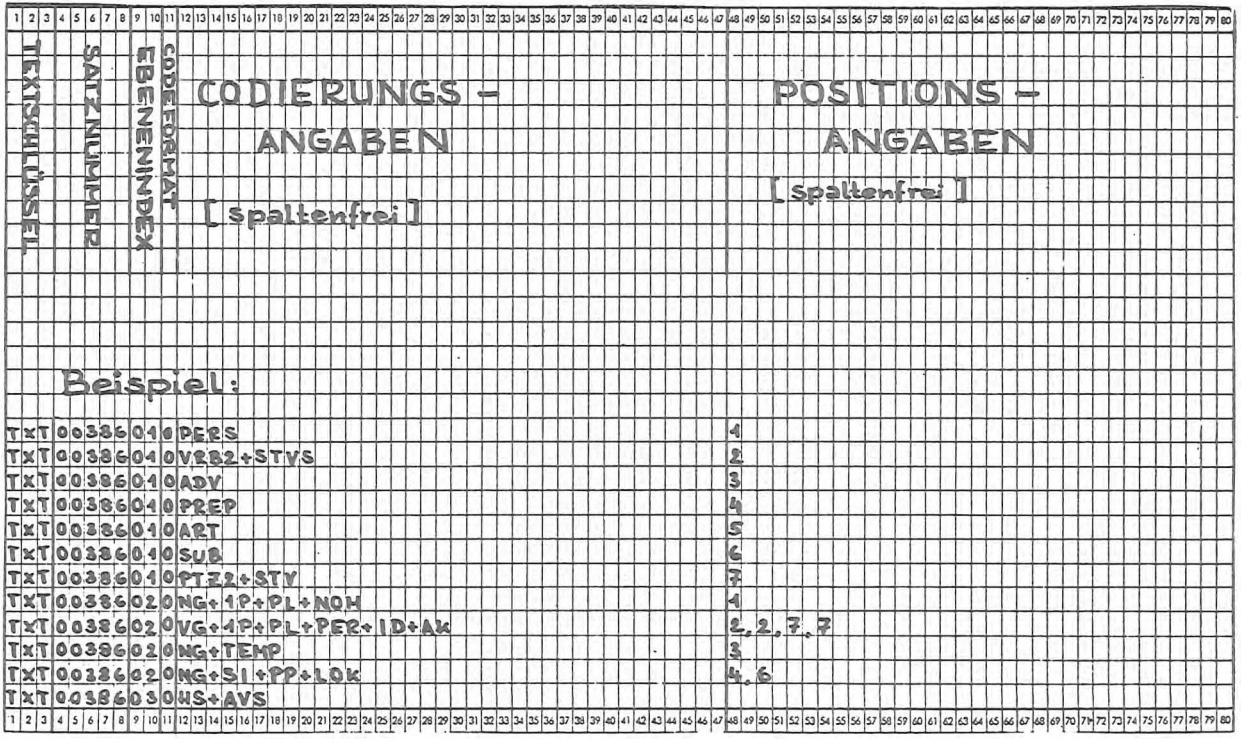
Die ausgefüllte Ablochunterlage wird nacheinander satzweise, innerhalb eines Satzes ebenenweise (also nacheinander der Inhalt der Spalten 3 bis 6 der Vorlage) abgelocht. Die einzelnen Sätze sind zur optischen Erleichterung durch stark hervortretende Kopfzeilen über den 6 Spalten voneinander getrennt.

Für das Lochen wird die Programmtrommel im Kartenlocher so programmiert (siehe auch Abbildung 33), daß die Spalten 1 bis 11 automatisch dupliziert werden, die Lochkarte anschließend auf Spalte 12 plaziert ist und durch Betätigen der "Sprung"-Taste als nächstes Ziel die Spalte 48 erreicht wird.

Mit Beginn eines neuen Satzes oder einer neuen Codierungsebene (neue Spalte) wird die Programmtrommel ausgeschaltet und die Spalten 1 bis 11 werden gelocht, und zwar in

1 bis 8 die Angaben, die in der Kopfzeile über der ersten Spalte des Maschinenausdrucks enthalten sind

Abbildung 32



(Textschlüssel und Satznummer),

9 bis 10 der Ebenenindex, der in der Kopfzeile über der Formularspalte, deren Inhalt zum Ablochen ansteht, handschriftlich eingetragen ist und einen der Werte 01, 02, 03, ... 10 hat,

11 das Codeformat, das unter dem Ebenenindex handschriftlich eingetragen ist (0 oder 1).

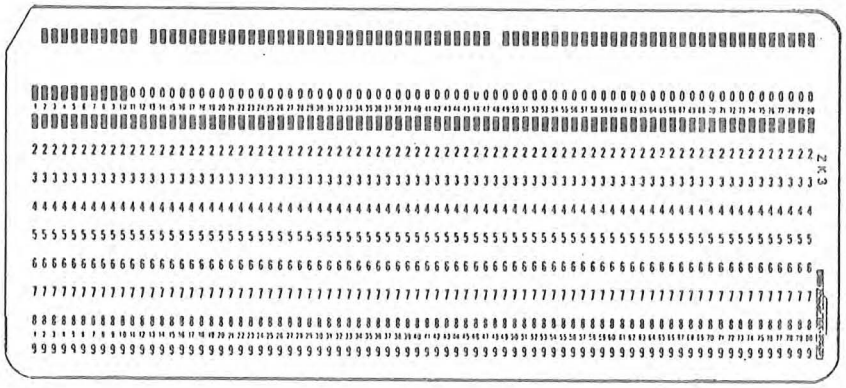
Danach wird die Programmtrommel wieder eingeschaltet und die ersten Codierungs- und Positionsangaben werden in den dafür vorgesehenen Feldern abgelocht, wobei das Feld für die Positionsangaben durch Drücken der "Sprung"-Taste erreicht wird.

Nun wird die Spalte der Vorlage, aus der die Angaben über Codierungsebene und Codeformat entnommen sind, sukzessive abgearbeitet, wobei immer die Spalten 1 bis 11 automatisch dupliziert werden. Die Bearbeitung der Spalte der Vorlage endet dann, wenn man auf eine neue Kopfzeile, also auf einen neuen Satz trifft.

Sind alle Codierungen des bearbeiteten Satzes erfaßt, also alle Spalten der Vorlage abgearbeitet, kann der folgende Satz wie beschrieben in Angriff genommen werden. Andernfalls wird an den Anfang des bereits bearbeiteten Satzes zurückgesprungen und die nächste Spalte für diesen Satz, die sich jetzt auf eine andere Codierungsebene bezieht, verfolgt. Dazu wird die Programmtrommel wieder abgeschaltet, der Inhalt der Lochkartenspalten 1 bis 11 wird neu erstellt mit den für die Spalte zutreffenden Angaben, die Programmtrommel wird wieder eingeschaltet und die folgenden Coderungs- und Positionsangaben werden erfaßt.

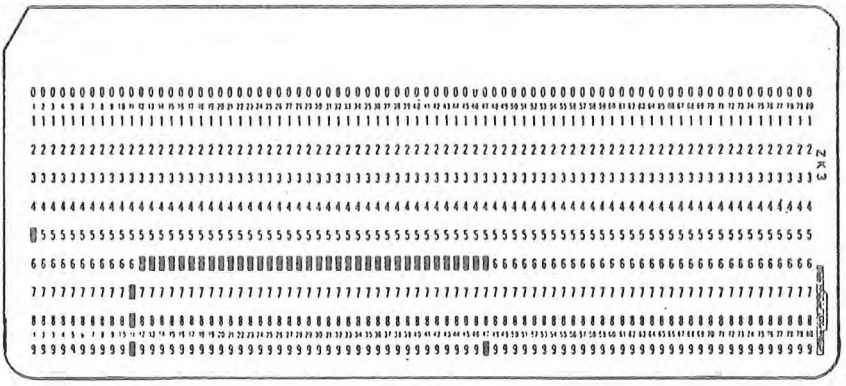
Abbildung 33

SCHREIBLOCHER IBM 029



SCHREIBLOCHER SIEMENS 2080-30

BULL 012



Ist an einer Stelle zu den Codierungs- und Positionsangaben zusätzlich noch ein Codeformat angegeben, das von dem in der Kopfzeile abweicht, wird folgendermaßen verfahren:

- Programmtrommel ausschalten
- Spalten 1 bis 10 von Hand duplizieren
- neues Codeformat (0 oder 1) lochen
- Codierungs- und Positionsangaben in den vorgesehenen Feldern lochen
- auf folgender Karte Spalten 1 bis 10 von Hand duplizieren
- altes Codeformat wieder eintragen
- Programmtrommel wieder einschalten

3. CODIEREN MIT DER TEXTVORLAGE AUS PC-22

PC-22 erstellt eine Textvorlage, in der die Wörter der einzelnen Sätze eines Textes durch Indizierung numeriert sind (siehe Abbildung 7). Diese Vorlage wird verwendet, wenn nur auf einer Ebene codiert wird.

Jeder Satz wird mit einer Überschrift eingeleitet, in der bereits Textschlüssel und Satznummer so abgedruckt sind, wie sie auf die zu erstellenden Lochkarten übernommen werden. Diese Überschrift wird handschriftlich ergänzt durch den Ebenenindex (2 Stellen) und das zutreffende Codeformat (1 Stelle).

Die Codierungs- und Positionsangaben, die nun bei der manuellen Textanalyse anfallen, werden gleich in ein Ablochschema für Daten (siehe Ab-

bildung 32) übernommen. Mit Beginn eines neuen Satzes werden die Spalten 1 bis 11 mit den Angaben in der Überschrift ausgefüllt; es folgen dann nur noch Codierungs- und Positionsangaben zum Satz. Erst für einen nächsten Satz wird die Überschrift wieder neu übernommen.

Aus diesem Ablochschema kann dann wieder in der bereits beschriebenen Weise abgelocht werden.

4. DIE BEARBEITUNG DES ERGEBNISPROTOKOLLS VON PC-30

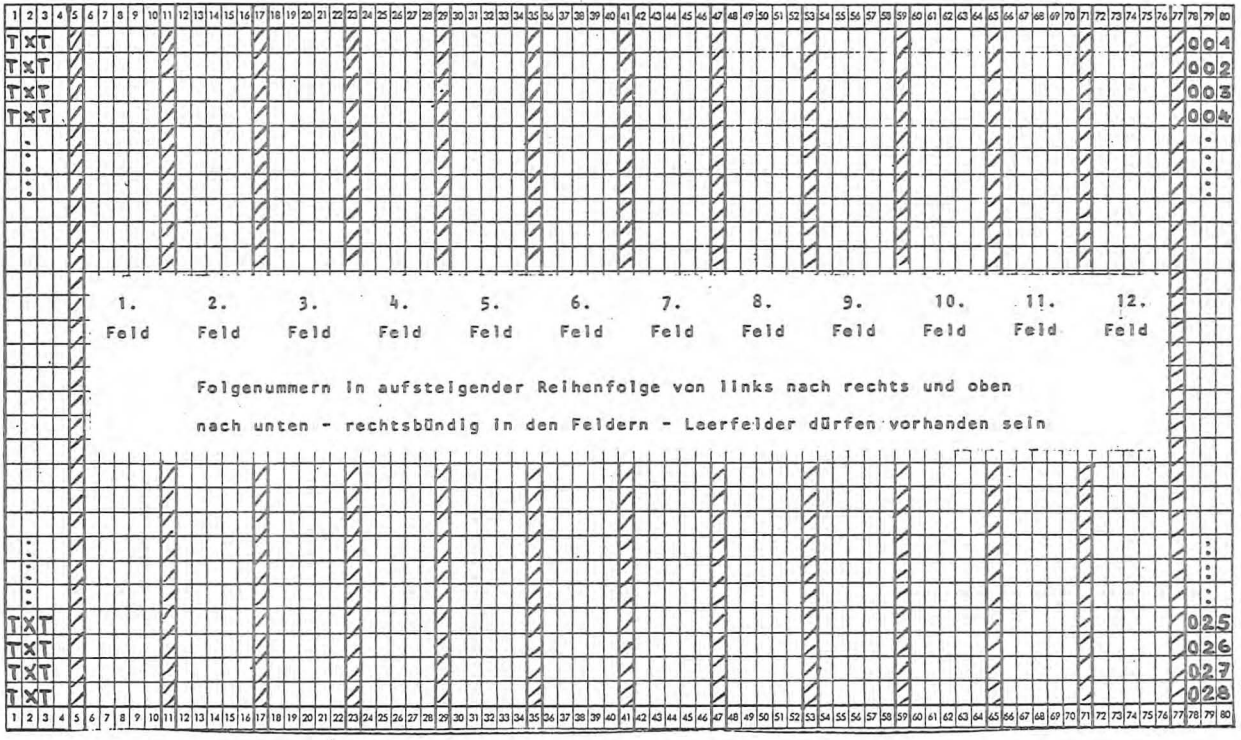
Das Programm PC-30 liefert als Ausgabeprotokoll eine Auflistung aller Codierungen (siehe Abbildungen 12 und 13), die bearbeitet wurden. Die Reihenfolge dabei ist die der Eingabe. Das Protokoll wird nun mit zwei Korrekturlisten zugleich bearbeitet.

Die eine Liste ist ein Ablochschema mit einer Spalteneinteilung, wie sie aus Abbildung 34 ersichtlich ist. In dieser Liste werden die Folgenummern der Codierungen gesammelt, die inhaltliche Fehler aufweisen. Diese Liste wird zeilenweise ausgefüllt. Eine Zeile kann bis zu 12 dieser Nummern aufnehmen. Die einzelnen Nummern müssen *rechtsbündig* in die vorgesehenen Felder eingetragen werden, Leerfelder dürfen vorhanden sein. Es müssen die einzelnen Nummern in aufsteigender Reihenfolge angegeben sein. Eine Folgenummer, die dieser Anforderung nicht genügt, wird mit der Meldung

SORTIERFEHLER - NICHT BEARBEITET

vom Programm PC-40 abgewiesen, die zugehörige Co-

Abbildung 34



dierung aber, obwohl inhaltlich falsch, in STMFIL übernommen. Die übrigen Codierungen, die durch ihre Folgenummern ausgewiesen sind und gelöscht werden, werden decodiert als Folge mnemonischer Kurzwörter abgedruckt, so daß eine endgültige Kontrolle des ordnungsgemäßen Ablaufs möglich ist. Die Spalten 1 bis 3 der Liste können den Textschlüssel aufnehmen; in den Spalten 78 bis 80 sollten die einzelnen Zeilen fortlaufend numeriert werden. Beide Felder werden jedoch vom Programm nicht gelesen.

Die zweite Liste, die bei der Korrektur des Ausdrucks geführt wird, ist ein Ablochschema für Codierungen (Abbildung 32). In dieses Ablochschema werden die als formal falsch zurückgewiesenen Codierungen jetzt korrigiert übernommen, ebenso, wenn zutreffend, die Codierungen, die inhaltliche Fehler aufwiesen. In diese Liste können zusätzlich neue Codierungen aufgenommen werden.

Zweckmäßigerweise wird das Protokoll nicht stückweise, sondern am Anfang beginnend bis zum Ende bearbeitet. Dann erhält man die richtige Anordnung der Folgenummern und es ist kein zeitaufwendiges Spulen der Magnetbänder bei der Übernahme der verbesserten Codierungen nötig.

Die Lochkarten, auf denen schließlich die Folgenummern der zu entnehmenden Codierungen abgelocht sind, werden dem Programm PC-40 zugeführt. Sind keine Streichungen nötig, muß PC-40 dennoch gestartet werden, dann allerdings ohne Datenkarten. Erst anschließend können durch PC-30 die Codierungen aus der zweiten Liste, sowie nach Belieben neue Codierungen aufgenommen werden (siehe Abbildung 23).

ANHANG II

Übersicht über das Programmsystem

PC-10: ERSTELLEN EINES CODEUMSETZERS

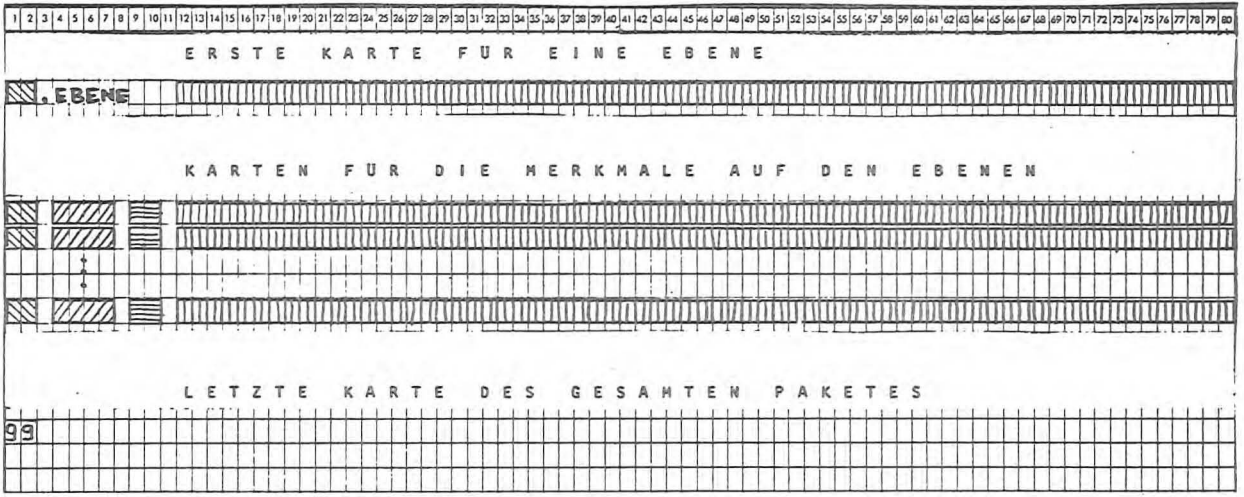
Die Eingabe für dieses Programm erfolgt über Datenkarten (Ablochschema siehe Abbildung 35).

Die Merkmale der einzelnen Ebenen, benannt durch mnemonische Kurzwörter, werden klassifiziert. Der Klassenindex 0 weist auf die Menge der obligatorischen Merkmale einer Ebene hin, die Klassen fakultativer Merkmale werden mit 1 beginnend fortlaufend numeriert. Die Eingabe erfolgt ebenenweise. Jede Ebene wird eingeleitet durch eine Datenkarte, die in den Spalten 1 und 2 den Ebenenindex (rechtsbündig) enthält. Ab Spalte 12 (bis 80) folgt dann ein beliebiger Kommentar, der die Ebene beschreibt. Darauf folgen die Karten zu der Ebene. Jede solche Karte enthält Ebenenindex, mnemonisches Kurzwort des Merkmals und Klassenindex, sowie eine Beschreibung des Merkmals als Kommentar.

Spalten	01 - 02	Ebenenindex (rechtsbündig)
	04 - 07	mnemonisches Kurzwort (linksbündig)
	09 - 10	Klassenindex (rechtsbündig)
	12 - 80	beliebiger Kommentar

Die Datenkarten sind ebenenweise, innerhalb einer Ebene klassenweise (aufsteigender Klassenindex) zusammengefaßt. Die letzte Karte des Datenkartenpakets enthält in Spalte 1 und 2 das Zeichen "9".

Abbildung 35



PC-11: ERWEITERN EINES CODEUMSETZERS

Das Erweitern eines Codeumsetzers bezieht sich auf das Einbringen von Merkmalen zu noch nicht erklärten Ebenen in den Codeumsetzer. Die Merkmale zu diesen Ebenen werden, wie oben zusammengefaßt und klassifiziert, in gleicher Weise wie bei PC-10 als Datenkarten eingegeben. Die letzte Karte enthält wieder in den Spalten 1 und 2 die Zeichen "99".

PC-12: VERÄNDERN EINES CODEUMSETZERS

Alle Merkmale, die im neuen Codeumsetzer enthalten sein sollen, werden in gleicher Weise wie bei PC-10 auf Datenkarten erfaßt und eingegeben. Für analoge Ebenen muß die Ebenenindizierung erhalten bleiben. Auch Ebenen, in deren Merkmalsvorrat nicht verändernd eingegriffen wird, müssen durch ihre Merkmale neu definiert werden. Der Aufbau des sich ergebenden Datenkartenpaketes ist dann der gleiche wie bei PC-10.

PC-20: ABLOCHUNTERLAGEN VON SATZZERLEGTEM TEXT

Bearbeitet wird ein Text, der als einziger File auf einem Magnetband gespeichert ist. Er wird als Codierungsvorlage aufbereitet wie dies aus Abbildung 6 ersichtlich ist. Es besteht die Möglichkeit, nur Teile des Textes aufbereiten zu lassen. Diese Teile werden durch die entsprechenden Satznummern eingegrenzt. Die Eingabe dieser Eingrenzung erfolgt über Datenkarten, wo-

bei jedes Teilstück auf einer eigenen Karte beschrieben wird: Sie enthält in den Spalten

01 - 05 Nummer des ersten Satzes des Teilstückes

06 - 10 Nummer des zweiten Satzes des Teilstückes

jeweils rechtsbündig. Soll der ganze Text aufbereitet werden, wird eine leere Datenkarte benötigt.

PC-21: ABLOCHUNTERLAGEN VON SATZZERLEGTEM TEXT (FKZ-BAND)

Dem Operator wird mitgeteilt, zu welchen Texten Codierungsvorlagen erstellt werden sollen. Von ihm werden die in Frage kommenden Magnetbänder ausgewählt. Über eine Anfrage an den Operator wählt das Programm einen Text aus und bereitet ihn auf (siehe Abbildung 6). Es besteht dabei nicht die Möglichkeit, nur Teile eines Textes bearbeiten zu lassen.

PC-22: WORTNUMERIERTE TEXTAUSGABE

Bearbeitet wird ein vollständiger Text, wobei es gleichgültig ist, ob ein Magnetband einen oder mehrere Texte enthält. Sind mehrere Texte enthalten, muß die Lage des gewünschten Textes innerhalb des Magnetbandes bekannt sein. Der zutreffende File muß durch zutreffende Positionierung des Bandes durch den Operator ausgewählt werden.

PC-30: KONTROLLE UND UMWANDLUNGEN VON CODIERUNGEN

Eingabe für dieses Programm sind Codierungen, wie sie nach einer Übertragung auf Lochkarten aus einer Codierungsvorlage oder dem entsprechenden Ablochschema (Abbildung 32) erstellt wurden. Die Codierungen sollen dabei satzweise, innerhalb eines Satzes ebenenweise in beliebiger Reihenfolge zusammengefaßt sein, um so unnötiges und zeitraubendes Spulen der Magnetbänder zu vermeiden. Die Anzahl der Codierungen ist beliebig; sie müssen aber alle zum gleichen Text gehören.

In der ersten Bearbeitungsstufe werden die Lochkarten eingelesen, die Felder mit Satznummern, Ebenenindex und Codeformat werden auf numerische Lochung geprüft, gegebenenfalls wird bei fehlerhafter Lochung für alle drei Werte O substituiert. Anschließend werden die Karten auf einem Magnetband (Arbeitsband) gespeichert.

In der zweiten Bearbeitungsstufe erfolgt die eigentliche Interpretation des Inhalts der einzelnen Karten: formal fehlerfreie Karten werden codiert, das Ergebnis dieser Umwandlung wird auf GENFIL festgehalten. Karten mit formalen Fehlern werden mit einer entsprechenden Fehlermeldung verbunden. Es wird ein Ausgabeprotokoll gefertigt, das alle Codierungen, soweit sie interpretiert werden konnten, numeriert wiedergibt und auch die fehlerhaften Karten mit der Fehlerkennung enthält.

PC-40: LOESCHEN UND INVENTARISIEREN VON CODIERUNGEN

GENFIL ist ein Zwischenspeicher und enthält interpretierte Codierungen in der Reihenfolge, wie sie im Ausgabeprotokoll von PC-30 durch Folgenummern aufgezeigt ist. PC-40 übernimmt diese Codierungen und mischt sie in den STMFIL zum Text. Codierungen, die bei dieser Übertragung nicht berücksichtigt werden sollen, weil sie inhaltliche Fehler aufweisen, sind durch ihre Folgenummern angegeben. Die Eingabe dieser Nummern erfolgt über Datenkarten, wie sie nach dem Ablochschema in Abbildung 34 erstellt wurden. Sollen alle Codierungen nach STMFIL übernommen werden, entfallen Datenkarten. Für entnommene Codierungen wird ein Protokoll geliefert, welches diese Codierungen als Folge mnemonischer Kurzwörter wiedergibt. Nach Beendigung der Übernahme von Codierungen nach STMFIL ist GENFIL für weitere Codierungen (PC-30) offen.

PC-50: DECODIEREN UMGEWANDELTER CODIERUNGEN

Das Programm kann angewendet werden auf GENFIL oder STMFIL mit der zugehörigen Text-Datei und liefert einen Überblick über alle dem jeweiligen Band vorhandenen Codierungen (siehe dazu Abbildung 24).

PC-51: DECODIEREN UMGEWANDELTER CODIERUNGEN AUF AUSGEWÄHLTEN EBENEN

Das Programm liefert einen Überblick über alle

auf GENFIL oder STMFIL vorhandenen Codierungen in gleicher Weise wie PC-50, allerdings nur auf den gewünschten Ebenen, aber unter Beibehaltung der Gesamtnumerierung. Die gewünschten Ebenen werden über eine Datenkarte mit folgendem Aufbau eingegeben (beginnend in Spalte 1, "_" bedeutet ein Leerzeichen, XX ist Index einer gewünschten Codierungsebene, rechtsbündig angegeben):

EBENENAUSWAHL:_XX/XX/XX/XX/XX/XX/XX/XX/XX/XX

Wieviele Ebenen in dieser Weise durch ihre Ebenenindizes angegeben werden ist beliebig, es können jedoch höchstens 10 sein.

PC-60: UMCODIEREN FÜR NEUEN CODEUMSETZER

Das Programm codiert einen Datenbestand auf STMFIL zu einem Text um. Die Decodierung der Merkmale erfolgt über den alten Codeumsetzer, der in einer Systemdatei reserviert ist. Die neue Codierung resultiert aus dem neuen Codeumsetzer, wie er mit PC-12 erstellt wurde. Es muß angegeben werden, auf welchen Ebenen umcodiert werden soll. Diese Ebenen werden wieder durch ihre Ebenenindizes angegeben, die Eingabe erfolgt über eine Datenkarte mit folgendem Aufbau (beginnend in Spalte 1, "_" bedeutet ein Leerzeichen, XX enthält rechtsbündig einen Ebenenindex):

UMCODIEREN_AUF:_XX/XX/XX/XX/XX/XX/XX/XX/XX/XX

Für die Anzahl n der so spezifizierten Ebenen muß $1 \leq n \leq 10$ gelten.

PC-70: SATZ-ORIENTIERTES RETRIEVAL

Durchsucht wird ein Text mit Hilfe der zugehörigen Codierungen auf STMFIL. Für die Suche werden über Lochkarten Suchbegriffe in beliebider Anzahl eingegeben. Der Aufbau eines Suchbegriffs ist aus Abbildung 25 ersichtlich.

PC-71: STATISTIK-ORIENTIERTES RETRIEVAL

Es gilt Gleiches wie bei PC-70.

PC-80: STRUKTURZEIGENDE TEXTAUFBEREITUNG

Die Codierungsebenen, zu denen decodiert werden soll, werden über Datenkarten angegeben. Zusätzlich wird für die einzelnen Codierungsebenen eine Klassenauswahl spezifiziert, die angibt, welche Klassen für die jeweilige Ebene berücksichtigt werden sollen. Die Klassenauswahl kann maximal 5 Klassen fakultativer Merkmale benennen, eine Decodierung der obligatorischen Merkmale für die Ebenen erfolgt in jedem Fall. Die Angaben zu den Ebenen werden auf drei Datenkarten mit folgendem Aufbau abgelocht (jeweils beginnend in Spalte 1, XX bedeutet einen Ebenenindex, YY gibt Klassenindizes an):

```
E1=XX(YY,YY,YY,YY,YY)
E2=XX(YY,YY,YY,YY,YY)
E3=XX(YY,YY,YY,YY,YY)
```

Die Indizes werden in ihren Feldern rechtsbündig abgelocht, die Reihe der Klassenindizes kann vorzeitig durch ")" beendet werden. Die Reihenfolge der Datenkarten zu den gewünschten Ebenen bestimmt die Reihenfolge, in der die Spalten schließlich im Ausdruck (siehe Abbildungen 30 und 31) aufeinander folgen. Maximal können 3 Ebenen spezifiziert werden, sind weniger Ebenen gewünscht, bleiben entsprechende Datenkarten leer.

A n m e r k u n g e n

[1] Siehe dazu:
Engel, Ulrich (Hrsg.): Forschungsberichte des Instituts für deutsche Sprache 3, 1968 und 4, 1969.

[2] Siehe beispielsweise die Arbeiten des Germanistischen Instituts und des Instituts für Angewandte Mathematik der Universität des Saarlandes.

[3] In:
Engel, Ulrich (Hrsg.): Forschungsberichte des Instituts für deutsche Sprache 2, 1968, Seite 30 ff.

[4] Genaueres siehe 3.1

[5] Zugrundegelegt wird ein Zeichenvorrat, wie er auf Datenerfassungsgeräten, beispielsweise einem Kartenlocher, vorhanden ist.

[6] Für die übrige Ausgabe siehe 4.2

[7] Unter welchen Bedingungen ein Umcodieren entfallen kann, ist in 5.4 dargestellt.

[8] Bauer - Goos: Informatik I, Berlin-Heidelberg-New York, 1971, Seite 28 ff.
Siehe auch: DIN 44 300

[9] Gross - Lentin: Mathematische Linguistik, Berlin-Heidelberg-New York, 1971, Seite 197

VERZEICHNIS DER ABBILDUNGEN

Seite

Seite

Seite

FORSCHUNGSBERICHTE DES INSTITUTS FÜR DEUTSCHE SPRACHE – MANNHEIM,
herausgegeben von Ulrich Engel und Irmgard Vogel

Band 1: I. Arbeitsberichte
Grundsätzliche Bemerkungen zu den Untersuchungen zum Verbalbereich
G.Beugel / U.Suida, Perfekt und Präteritum in der deutschen Sprache der Gegenwart
H. Gelhaus, Das Futur in der deutschen Sprache
S. Jäger, Zum Gebrauch des Konjunktivs in der indirekten Rede
K. Brinker, Das Passiv
II. Diskussionsbeiträge
S. Jäger, Der Modusgebrauch in den sogenannten irrealen Vergleichssätzen
B. Engelen, Zur Semantik des deutschen Verbs
U. Engel, Adjungierte Adverbialia. Zur Gliederung im Innenfeld
Mannheim 1968, 103 S., DM 8.–. Tübingen [2] 1972
ISBN 3–87808–601–6

Band 2: U. Engel, Vorbemerkungen
I. Zint, Maschinelle Sprachbearbeitung des Instituts für deutsche Sprache in Mannheim (Teil I)
M. W. Hellmann, Zur Dokumentation und maschinellen Bearbeitung von Zeitungstexten in der Außenstelle Bonn (Teil II)
G. Billmeier, Über die Signifikanz von Auswahltexten (Teil III)
Mannheim 1968, 171 S., DM 8.–. Tübignen [2] 1973
ISBN 3–87808–602–4

Band 3: P. Kern, Bemerkungen zum Problem der Textklassifikation
M. W. Hellmann, Über Corpusgewinnung und Dokumentation im Mannheimer Institut für deutsche Sprache
W. Müller, Teilerhebungen und ihre Anwendung auf die Sprachbearbeitung
U. Engel, Das Mannheimer Corpus
Mannheim 1969, 84 S., DM 8.–
ISBN 3–87808–603–2

Band 4: B. Engelen, Das Präpositionalobjekt im Deutschen und seine Entsprechungen im Englischen, Französischen und Russischen
M. H. Folsom, Zwei Arten von erweiterbaren Richtungsergänzungen
A. Ströbl, Aus den Überlegungen zur Bearbeitung der Wortstellung für das „Grunddeutsch"
Ch. Winkler, Untersuchungen zur Intonation in der Deutschen Gegenwartssprache
R. M. Frumkina, Über das sogenannte „Zipfsche Gesetz". (Aus dem Russischen übersetzt von A. Schubert)
Mannheim 1970, 132 S., DM 12.60. Tübingen [2] 1974
ISBN 3–87808–604–0

Band 5: U. Engel, Regeln zur Wortstellung
U. Winkelstern, Corpusanalyse zur Untersuchung der Wortstellung
B. Busch, Erfahrung bei der Codierung
Mannheim 1970, 170 S., DM 8.–
ISBN 3–87808–605–9

Band 6: Sammelband
B. Engelen, Referentielle und kontextuelle Determination des Wortinhaltes als Problem der Wortarten
H. Fenske, Zur Verschlüsselung von Satzbauplänen. Ein Arbeitsbericht
S. Jäger, Hochsprache und Sprachnorm. Kritische Bemerkungen zu einer sprachwissenschaftlichen Verfahrensweise
Tübingen 1971, 100 S., DM 8.–
ISBN 3–87808–606–7

Band 7: Gesprochene Sprache. Bericht der Forschungsstelle Freiburg
Tübingen 1973, 311 S., Dreifachband DM 24.–
ISBN 3–87808–607–5

Band 8: S. Jäger, J. Huber, P. Schätzle, Sprache – Sprecher – Sprechen. Probleme im Bereich soziolinguistischer Theorie und Empirie
Tübingen 1972, 377 S., Dreifachband DM 24.–
ISBN 3–87808–608–3

Band 9: H. Popadić, Untersuchungen zur Frage der Nominalisierung des Verbalausdrucks im heutigen Zeitungsdeutsch
Tübingen 1972, 151 S., DM 8.–
ISBN 3–87808–609–1

Band 10: H. Fenske, Schweizerische und österreichische Besonderheiten in deutschen Wörterbüchern
Tübingen 1973, 390 S., Dreifachband DM 24.–
ISBN 3–87808–610–5

Band 11: I. Neumann, Temporale Subjunktionen. Syntaktische-semantische Beziehungen im heutigen Deutsch
Tübingen 1972, 180 S., DM 8.–
ISBN 3–87808–611–3

Band 12: G. Kaufmann, Das konjunktivische Bedingungsgefüge im heutigen Deutsch
Tübingen 1972, 168 S., DM 8.–
ISBN 3–87808–612–1

Band 13: P. Nikitopoulos, Statistik für Linguisten. Teil I
Tübingen 1973, 160 S., DM 12.60
ISBN 3–87808–613–X

Band 14: K. Bayer, K. Kurbel, B. Epp, Maschinelle Sprachbearbeitung im Institut für deutsche Sprache
Tübingen 1974, 210 S., DM 19.60
ISBN 3–87808–614–8

Band 15: H. Gelhaus, S. Latzel, Studien zum Tempusgebrauch im Deutschen
Tübingen 1974, 350 S., DM 28.–
ISBN 3–87808–615–6

Band 16: Horst Raabe (hg.), Trends in kontrastiver Linguistik I. Band I: Interimssprache und kontrastive Analyse. Das Zagreber Projekt zur angewandten Linguistik
Tübingen 1974, 229 S., DM 25.20
ISBN 3–87808–616–4

Band 17: Signe Marx-Nordin, Untersuchungen zur Methode und Praxis der Analyse aktueller Wortverwendungen. Aspekte des Gebrauchs der Wörter ‚Sozialismus' und ‚sozialistisch' in der politischen Sprache der DDR
Tübingen 1974, 229 S., DM 25.20
ISBN 3–87808–617–2

Band 18: Arbeitsgruppe MasA: Zur maschinellen Syntaxanalyse I. Morphosyntaktische Voraussetzungen für eine maschinelle Sprachanalyse des Deutschen
Tübingen 1974, 2 Teilbände zus. DM 64.–, 670 S.
ISBN 3–87808–618–0

Band 19: Arbeitsgruppe MasA: Zur maschinellen Syntaxanalyse II. Ein Lexikon für eine maschinelle Sprachanalyse des Deutschen
Tübingen 1974, 229 S., DM 25.20
ISBN 3–87808–619–9

Band 20: Heinz Kloss (hg.), Deutsch in der Begegnung mit anderen Sprachen: im Fremdsprachen-Wettbewerb, als Muttersprache in Übersee, als Bildungsbarriere für Gastarbeiter. Beiträge zur Soziologie der Sprachen
Tübingen 1974, 204 S., DM 19.60
ISBN 3–87808–620–2

Weitere Bände in Vorbereitung.

Stand 30. 9. 1974